Poste Italiane • Spedizione in A.P. 45% art. 2 comma 20/b legge 662/96 Aut. N. DCDC/033/01/CS/CAL

Anno III - Num. 16 • Febbraio 2002
Periodicità Mensile • € 4,90 • L. 9.488

ADSL Guida completa per la configurazione

ATTUALITÀ

Alla scoperta di FreeNet, la nuova rete peer-to-peer

Desktop 2002.... Cosa c'è di nuovo in KDE e GNOME?

TECNICHE & PROGRAMMAZIONE

Java Server Pages.... primi passi

Analisi comparativa dei JFS

APPROFONDIMENTI

A lezione con il PostgreSQL

Lotta al Cybercrime

COVER STORY

I "cugini" di Linux: gli Unix commerciali

Linux VS Windows XP! Chi la spunta?

TUTORIAL

Satan – Alla ricerca degli intrusi nella nostra rete

Destreggiarsi tra Log e file di configurazione

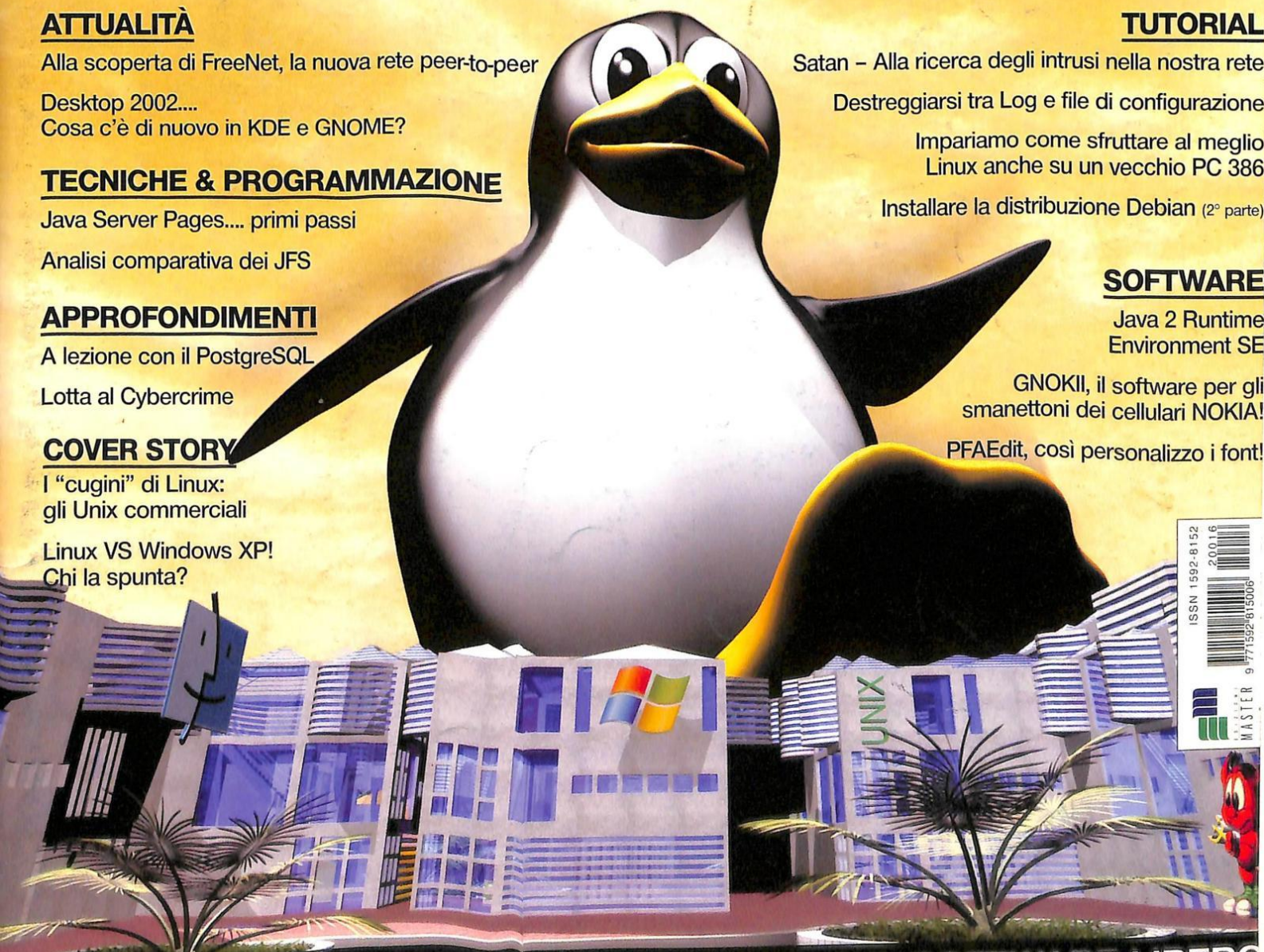
Impariamo come sfruttare al meglio Linux anche su un vecchio PC 386

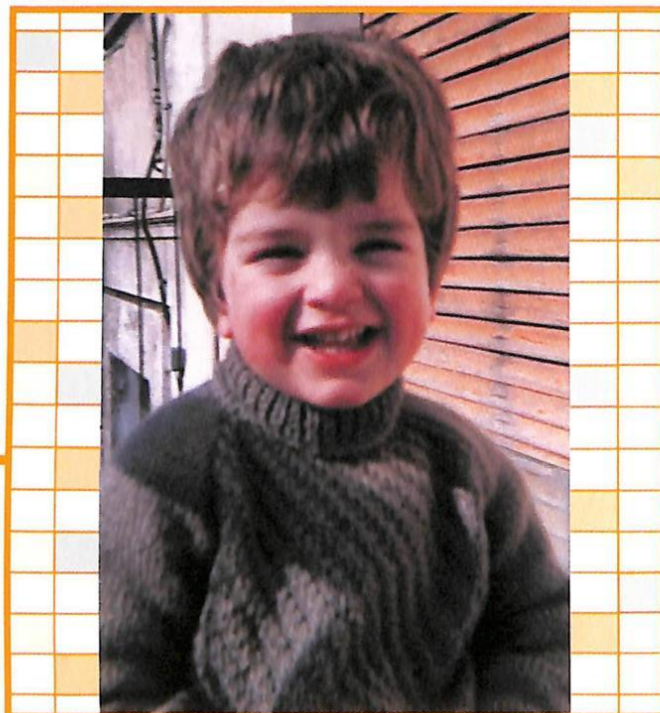
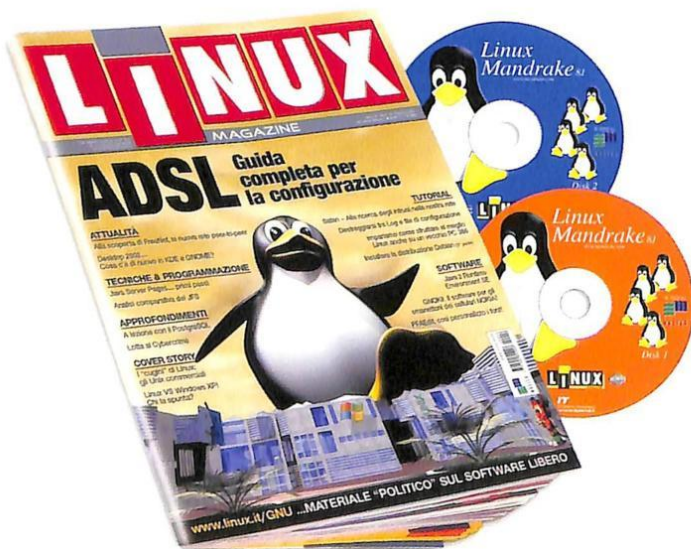
Installare la distribuzione Debian (2° parte)

SOFTWARE

Java 2 Runtime Environment SE

GNOKII, il software per gli smanettoni dei cellulari NOKIA!

PFAEdit, così personalizzo i font!

ISSN 1592-8152

www.linux.it/GNU ...MATERIALE "POLITICO" SUL SOFTWARE LIBERO

Non ho visto Harry Potter - il film per bambini strapompato l'ultimo Natale - non lo guarderò io e non incentiverò mia figlia, o qualsivoglia altro, ad occuparsene. Spero - ma dubito- che cada velocemente nell'oblio piuttosto che diventare una moda degli infanti telev-idioti. Non ho visto Codice Swordfish - il film con John Travolta - e non lo guarderò. E fortunatamente questo nell'oblio c'è già finito. In generale il cinema, tranne qualche classico americano ormai ben stagionato, m'attira poco. In tutti i film hollywoodiani la separazione tra il bene e il male è così netta e lampante da non lasciar dubbi all'interpretazione. Ma in questi ultimi film -e in gran parte dei nuovi prodotti dell'editoria di Hollywood- è ormai rintracciabile una comune e costante impronta distintiva: il nuovo cattivo è sempre sommariamente identificato con un hacker. Infatti, dopo una prima (leggera) infatuazione per il mondo che aveva generato questi giovani talenti della programmazione, nella "fabbrica dei sogni" queste rappresentazioni hanno quasi subito lasciato il posto alla più squallida caricatura rappresentata dall'infinita serie dei Nerds, dall'immagine del perenne adolescente brufoloso, monomaniaco, autoerotomane, asociale, inumano. Un'immagine paradossalmente errata, almeno a considerare la vita reale e il loro successo economico, la capacità di crearsi reti di relazioni e di contatti, ben superiore a qualunque altro. Una immagine che però è stata ben fissata nell'immaginario collettivo. Oggi questa tendenza è passata. Ma ci saremmo aspettati una maturazione tanto dei giovani brufolosi quanto degli sceneggiatori. Ma la storia sta andando diversamente. La tecnologia, rappresentata non casualmente nei film ormai solo dalla più spinta gadgettistica di marca, non ha più nulla di umano in questi film. C'è sempre un dispositivo -acquistabile poi in ogni negozio di elettrodomestici- in grado di risolvere qualunque problema, dietro il quale c'è solo un marchio multinazionale, e mai un progettista o un programmatore, un uomo. I tempi mitici di Wargames sono lontani. E quanto è diversa la meravigliosa Angelina Jolie di Hacker dall'irreale Lara Croft di quest'ultimo Natale. Se la condannata pubblicità "Copiare Software è Reato" della Business Software Alliance, rappresentava il sogno malcelato dei produttori di software commerciale di mandare in galera tutti i propri clienti se si fossero azzardati a trovare soluzioni alternative, che dire allora di questa nuova immagine degli hacker ammannita dal cinema? E se Hollywood è da sempre stata la fabbrica dei sogni possiamo subito capire come questa trasformazione rappresenti un vera e propria risposta all'incubo dei produttori. Il confronto tra la grande industria cinematografica, saldamente nelle mani di poche mastodontiche organizzazioni, è i giovani e talentuosi programmatori che hanno dimostrato di essere in grado di scardinare uno dopo l'altro tutti i tentativi di restringere i diritti di fruizione degli utenti -solo facendo appello alla propria umanissima capacità di fare, senza poter (e voler) fare appello a nessun gadget salvifico. Alle restizioni sulla riproduzione dei DVD, impossibili da riprodurre su piattaforme non decise dall'industria, il Free Software ha risposto con un programmino per Linux. Alle denunce, condanne e incarcerazioni pretese per questi giovani programmatori, per quanti hanno anche solo pubblicato le notizie e i link, la comunità ha risposto ridicolizzando l'estabilishment di Hollywood, fino ad arrivare a rappresentare l'algoritmo di decifrazione dei DVD con un numero primo - tanto per dimostrare quanto assurdo fossero quelle pretese di zittire questi "hacker", che in realtà stanno solo difendendo il proprio diritto di venir considerati proprietari di quanto comprano. Hollywood, che ha straperduto nel diritto, e non ha speranza nelle corti di giustizia ha cambiato strategia. Il cattivo malvagio in Harry Potter, è la caricatura del bambino prodigio con i computer, prototipo di questi giovani programmatori di talento, figlio di un programmatore di talento, che farà come tutti i cattivi una pessima fine. Se anche questo nulla volesse dire nell'economia generale di un mercato cinematografico che dalla fine della guerra fredda non sa proprio più dove cercarsi i nemici per catalizzare l'odio degli spettatori, lasciate che abbia appreso con molta diffidenza come il cattivo di Codice Swordfish, non sia altro che un giovane norvegese, genio della programmazione, arrestato all'arrivo negli USA con la solita operazione dell'FBI in assetto da guerra. E che il suo nome fosse... Torvalds, Axel Torvalds. "Se i sogni son desideri..."

IL SOGNO DI HOLLYWOOD: ARRESTARE TORVALDS, Linus Torvalds?

Emmanuele Somma
<esomma@ieee.org>

LINUX Magazine

Supplemento di ioProgrammo n° 55
Anno IV N.16 - Febbraio 2002
Periodicità Bimestrale
Reg. Trib. di CS n.ro 593/97
E-mail: linuxmag@edmaster.it
Internet: www.edmaster.it/linuxmag

Direttore Editoriale: Massimo Sesti
Direttore Responsabile: Romina Sesti

Coordinamento Red.: Emmanuele Somma
Redazione: Gianfranco Forlino
Collaboratori: A. Bennardi, G. Donvito, F. Ferroni, M. Giangrasso, F. Marasco, F. Marchetti-Stasi, S. Iovacchini, C. Stumpo, A. Pace, A. Pasqua, F. Tornieri, G. Tradigo, C. Vasselli, U. Zanatta

REALIZZAZIONE GRAFICA
CROMATIKA S.r.l.
Coord. grafico: Paolo Cristiano
Coord. Tecninco: Giancarlo Sicilia
Illustrazioni: Salvatore Vuono
Impaginazione elettronica: Aurelio Monaco

PUBBLICITÀ
Edizioni Master S.r.l.
Via Cesare Correnti, 1 - 20123 - Milano
Tel. 02 8321612 - Fax 02 8321764
email: advertising@edmaster.it

EDITORE
Edizioni Master S.r.l.
Sede di Milano: Via Cesare Correnti, 1 - 20123 Milano
Tel. 02 8321482 - Fax 02 8321699
Sede di Cosenza: C.da Lecco, zona ind. - 87030 Rende (CS)
Responsabile Amministrazione: Benedetto Celsa
Responsabile di Produzione: Nicolino Rocca
Diffusione: Giuseppina Bruno
Marketing: Antonio Meduri, Alessandra Fantozzi

Abbonamenti e arretrati: Costo abbonamento annuale (11 numeri): € 37.00 (L. 71.642). Costo arretrati (a copia): il doppio del prezzo di copertina + € 5,16 (L.10.000) spese (spedizione con corriere). (Prima di inviare i pagamenti, verificare la disponibilità delle copie arretrate al num. Telef. 028321482). La richiesta contenente i Vs. dati anagrafici e il nome della rivista, dovrà essere inviata via fax al num. 028321699, oppure via posta a:
EDIZIONI MASTER S.r.l. via Cesare Correnti, 1
20123 Milano, dopo avere effettuato il pagamento, secondo le modalità di seguito elencate:

- cc/p n.16821878 o vaglia postale (inviando copia della ricevuta del versamento insieme alla richiesta),
- assegno bancario non trasferibile (da inviarsi in busta chiusa insieme alla richiesta);
- carta di credito, circuito VISA, CARTASI', MASTERCARD/EUROCARD, (inviando la Vs. autorizzazione, il numero della carta, la data di scadenza e la Vs. sottoscrizione insieme alla richiesta).

SI PREGA DI UTILIZZARE IL MODULO RICHIESTA ABBONAMENTO POSTO NELLE PAGINE INTERNE DELLA RIVISTA. L'abbonamento verrà attivato sul primo numero utile, successivo alla data della richiesta.

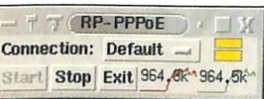
Sostituzioni: Inviare il Cd-Rom difettoso in busta chiusa a: Edizioni Master - Servizio Clienti
Via Cesare Correnti, 1 - 20123 Milano
Assistenza tecnica: linuxmag@edmaster.it

Servizio clienti:
☎ tel.02 8321482
@ e-mail:servizioclienti@edmaster.it

Stampa: Agem S.p.A. - (CL)
Stampa CD-Rom: Disctronics Italia S.p.A. (MI)
Distributore: Parrini & C. S.p.A. - ROMA

Finito di stampare: Gennaio 2002

La Edizioni Master edita:
Idea Web, Go!OnLine Internet Magazine, Win Magazine, Quale Computer, DVD Magazine, Office Magazine, ioProgrammo, Linux Magazine, Più Giochi PS2, Softline Software World, MPC, Maxi 3D Collection, Computer Games Gold, inDVD, I Fantastici CD-Rom, PC VideoGuide, Tutto Software, Go!OnLine Collection, Win Trio, Quale Computer Collection, DVD Collection, ioProgrammo Collection, Office Collection, Maxi 3D Super Collection, MPC Trio.

© Creativo@Work

Soluzioni infinite.

Dal bit al giga... questa la mole di lavoro che giorno dopo giorno la tua LAN deve essere in grado di gestire.

Una miriade di dati indispensabili che devono giungere a destinazione in modo semplice, veloce ed affidabile.

Digicom Networking Solutions, una vasta gamma di prodotti per rendere il tuo ufficio sempre più professionale ed innovativo.

Schede di rete PCI LAN Giga 10/100/1000, HUB e SWITCH da 5 a 24 porte Dual Speed, schede LAN PC Card, Printer Server.

Ed inoltre tecnologie Wireless-LAN e soluzioni xDSL.

Networking digicom... soluzioni infinite.

Networking Solutions

PCI LAN Giga 10/100/1000
- Scheda di rete PCI 64/32 Bit fino a 1 Ghz

Mini HUB 10/100
- Hub 8 porte Ethernet 10/100 BaseTX
- 1 porta up-link

Dual Speed SWITCH 8G
- Switch 8 porte Ethernet 10/100 BaseTX
- Full-Duplex

PCI LAN 10/100
- Scheda di rete 10/100
- Full Duplex Plug&Play
- Disponibile anche in formato PC CARD (Palladio LAN 10/100)

Printer Server 10/100
- Porta LAN 10/100
- Porta parallela

www.digicom.it

Numero Verde 800-407787

Il mondo nelle Tue mani.

Direttorato Gnome, sconfitta per RMS

Deludenti per il pioniere del software libero i risultati delle elezioni delle menti direttive del celebre ambiente desktop.

La sconfitta di Richard M. Stallman (RMS) sorprende pochi: le sue posizioni estremistiche non costituiscono la premessa ideale per un incarico di tipo "politico". "Ho la reputazione di una persona incapace di compromessi" dice Stallman nel rispondere alle domande rivolte a tutti i candidati, "ma questa reputazione proviene dalle persone che non concordano con me su questioni fondamentali relative alla libertà. Sono anche insistente sul fatto che al progetto GNU vengano riconosciuti i giusti meriti. Ma su questioni puramente pratiche ho anni di esperienza di patteggiamento." Stallman ha raggiunto la diciottesima posizione in graduatoria, raccogliendo cinquanta voti su un totale di 420. I posti disponibili per il direttorato erano undici. I commenti tipici possono definirsi eufemisticamente "sollevati": "Ho un gran rispetto per Stallman, sia come programmatore che come filosofo" ha detto un lettore di Linux Today; "detto questo, ringrazio Dio che non sia stato scelto per il direttorato di Gnome!".

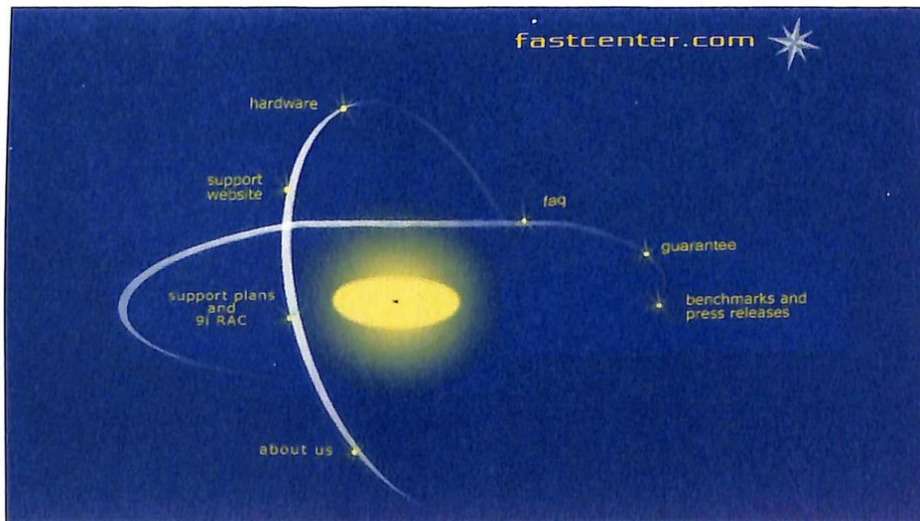
Per maggiori informazioni:
foundation.gnome.org

ORACLE, MEGLIO LINUX CHE WINDOWS

Le prestazioni su Windows 2000 server sono l'85% di quelle raggiunte con SuSE 7.

I benchmark sono stati effettuati da Fastcenter, una compagnia specializzata in vendita e supporto Oracle. Il test è stato effettuato con la versione 9.0.1 di Oracle, con lo scopo di verificare la differenza in termini di prestazioni tra Windows 2000 server e una distribuzione Linux, in questo caso la SuSE 7; pertanto, è stato utilizzato esattamente lo stesso hardware, e non sono state effettuate ottimizzazioni particolari rispetto alle installazioni di default.

I risultati sono stati nettamente in favore di SuSE. La costruzione di un database in condizioni di default ha impiegato 4 minuti e 12 secondi in SuSE , il 70% dei 5 minuti e 56 secondi impiegati da Windows 2000; il throughput medio di Windows 2000 è risultato essere l'85% di quello di SuSE.

Per maggiori informazioni:
www.fastcenter.com

Stabile o instabile, questo è il dilemma...

Parto tormentato per la nuova versione "instabile" di Linux.

Quasi un anno è stato necessario per il passaggio dalla versione 2.4 alla 2.5 del kernel di Linux. Per chi non lo sapesse, le versioni in cui la seconda parte del numero di versione è dispari sono quelle "di sviluppo", normalmente utilizzate per la realizzazione di nuove caratteristiche; invece nelle versioni "stabili", riconoscibili da un numero pari nella seconda parte del numero di versione, vengono solo corretti i bachi e inserite nuove caratteristiche di provata affidabilità.
Normalmente, tra il rilascio di una nuova versione "stabile" e la corrispondente versione "di sviluppo" passano poche settimane, utilizzate per risolvere i problemi più urgenti e avere così una base di partenza maggiormente affidabile sia per la versione di sviluppo che per quella di base. Tra la 2.4 e la 2.5, comunque, è passato quasi un anno, e gli avvenimenti di questo periodo non sono stati rassicuranti per quella parte dell'industria che tenta di utilizzare Linux come base per l'offerta di servizi, e che osserva con preoccupazione l'originale modello di sviluppo del kernel.
Già il rilascio della versione 2.4 era avvenuto con grande ritardo rispetto alle previsioni, e con un tono da parte di Linus Torvalds che lasciava perplessi: "Quel che è troppo è troppo", queste le parole di apertura dell'annuncio ufficiale. Correva gennaio 2001, e i mesi trascorsi hanno mostrato che probabilmente Linus non era soddisfatto di una componente fondamentale del nuovo kernel: il sottosistema della "memoria virtuale", che si occupa del cosiddetto processo di swapping , il trasferimento di dati tra memoria fisica (RAM) e disco, necessario quando la prima inizia a scarseggiare. Così, a metà di quest'anno Andrea Arcangeli, hacker italiano che lavora per la SuSE, si è assunto il gravoso onere di riscrivere da zero questo sottosistema.
Sfortunatamente, o forse per fortuna, tra le tante persone perplesse dalla sostituzione di una componente così importante in una versione "stabile" c'era anche Alan Cox, il "numero due" dello sviluppo di Linux, responsabile ufficiale della manutenzione della versione stabile. Cox ha deciso di mantenere un ramo distinto dello sviluppo del kernel, con una numerazione analoga a quella del ramo ufficiale e distinta per l'aggiunta di una sigla finale "ac", che continuava a utilizzare il precedente sistema di memoria virtuale, sviluppato da Rik van Riel. A tutti gli effetti, il tanto temuto forking di Linux ha avuto luogo, almeno per un periodo. Alla fine di tutto questo ambaradan, Linus si è detto soddisfatto della nuova memoria virtuale, e ha rilasciato la versione 2.5.0. Per la manutenzione del ramo "stabile" 2.4.x è stato scelto il diciottenne brasiliano Marcelo Tosatti, secondo indicazione di Alan Cox e con il benestare di Linus; la serie 2.4.x-ac sarà probabilmente destinata all'oblio, ma è verosimile che continuerà ad essere utilizzata per un po' di mesi. Con questo lieto fine, tutta la storia può essere rivista nella semplice ottica di un'erronea numerazione delle versioni: per rimediare a un rilascio prematuro della 2.4, Linus ha iniziato uno sviluppo "da numero dispari" nel bel mezzo di una versione stabile, in modo da potervi inserire una memoria virtuale di cui fosse soddisfatto. Il fine è senz'altro lieto per gli appassionati singoli e smaliziati, ma lascia spazio ad ampie preoccupazioni da parte dei produttori informatici che utilizzano Linux.

Marcelo, o Pinguim Maravilhoso

"É isso. Tornar-se um mantenedor do kernel repentinamente aumenta seu apelo sexual 1000% por alguma razão mágica."
- Linus

O jovem nas fotos acima é Marcelo Tosatti, o novo mantenedor do kernel 2.4 do Linux. Linus Torvalds e Alan Cox escolheram Marcelo por causa da sua experiência em programação do kernel, sua habilidade em comunicar-se com todos os desenvolvedores e sua alta dose de bom senso.

Per maggiori informazioni:
lwn.net/2001/1115
kt.zork.net/kernel-traffic/kt20011203_144.html
www.linuxtoday.com/news_story.php3?ltsn=2001-11-23-002-20-NW-KN

SICUREZZA NELLA SEGRETEZZA?

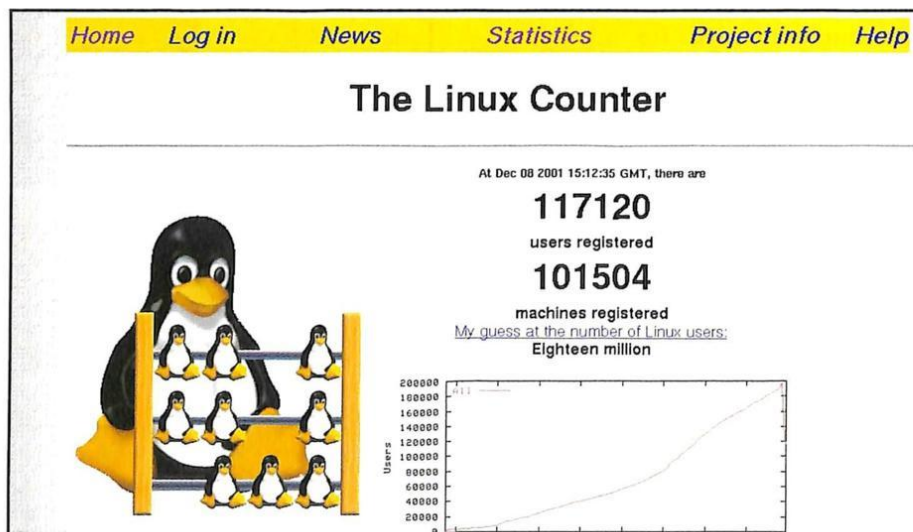
Nuova polemica tra la Microsoft e la comunità degli esperti di sicurezza informatica.

I virus Code Red e Nimda non si sono limitati a danneggiare migliaia di siti e milioni di utenti di computer, hanno anche inferto un duro colpo all'immagine della Microsoft, che si è ritrovata a fronteggiare dure critiche persino da un tradizionale alleato come la società di analisi Gartner Group.
La risposta non ha tardato ad arrivare. Scott Culp, responsabile del ``Microsoft Security Response Center", condanna in un suo articolo la comune pratica della "full disclosure", definendola "anarchia dell'informazione".
Per gli esperti di sicurezza la "full disclosure", cioè la pubblicazione dettagliata dei problemi di sicurezza del software, con tanto di programmi che la mettono in chiara evidenza, rappresenta uno strumento lavorativo di base. Rispetto ai classici comunicati stampa Microsoft, l'articolo di Pulp è equilibrato e ben ragionato; fortunatamente, la risposta, a difesa della libera circolazione dell'informazione, è arrivata da uno dei più celebri esperti di sicurezza informatici al mondo, Bruce Schneier, dalle pagine della sua rivista virtuale "Crypto Gram".
"Negli ultimi otto anni" dice Schneier, "abbiamo imparato che la full disclosure aiuta molto più di quanto non danneggi". In una ragionata difesa della libera circolazione dell'informazione, Schneier spiega che solo questa è in grado di costringere i produttori di software a risolvere i bachi di sicurezza anziché ignorarli e tentare di tenerli nascosti; ed è solo la libera circolazione dell'informazione a consentire a chiunque di prendere decisioni informate sulla sicurezza informatica. Come potrebbe essere, aggiungiamo noi, quella di non utilizzare prodotti bacati come quelli della Microsoft.

Per maggiori informazioni:
www.counterpane.com/crypto-gram-0111.html
www.microsoft.com/technet/columns/security/noarch.asp

Pinguini di tutto il mondo, contatevi!

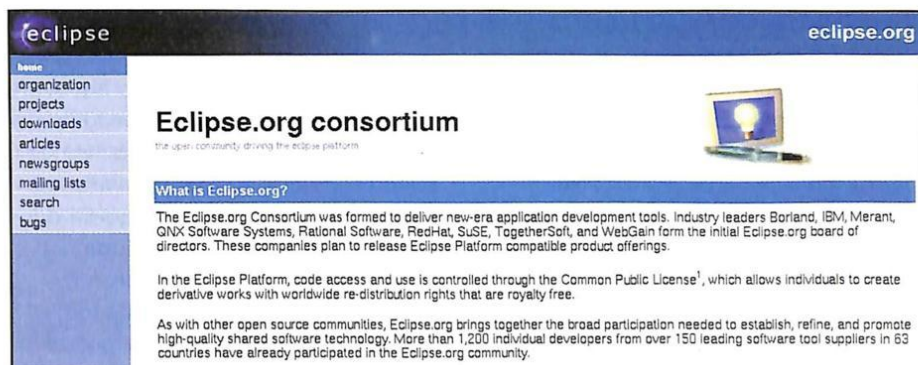
I nuovi dati del Linux Counter danno interessanti informazioni sulla diffusione delle diverse distribuzioni.

Non è la prima volta che parliamo del "Linux Counter", l'unico progetto di raccolta di dati sugli utenti Linux. A novembre è stata effettuata un'operazione di "pulizia" dei dati, cancellando gli utenti non piú raggiungibili via posta elettronica, e per l'occasione sono state aggiornate le varie statistiche disponibili. I risultati piú interessanti sono probabilmente quelli sulla popolarità delle distribuzioni: Red Hat 27.99%, Debian 21.11%, Mandrake 16.53%, Slackware 14.61%, SuSE 9.28%. Quasi contemporaneamente, il "Mandrake Forum" ha pubblicato un interessante articolo in cui vengono discussi e commentati i diversi metodi usati per effettuare statistiche di questo tipo; la conclusione piú importante è che tutti portano praticamente alle stesse quattro distribuzioni dominanti, vale a dire Debian, Mandrake, Red Hat e SuSE.

(Fa eccezione la Debian relativamente ai dati di vendita, che per una distribuzione non commerciale non sono disponibili.) Il dato è soprattutto interessante proprio per la Mandrake, che in un tempo molto breve è riuscita a guadagnare un'importante fetta di mercato.

Per maggiori informazioni:
counter.li.org
www.mandrakeforum.com/article.php?sid=1407

Inizia l'Eclisse del software proprietario

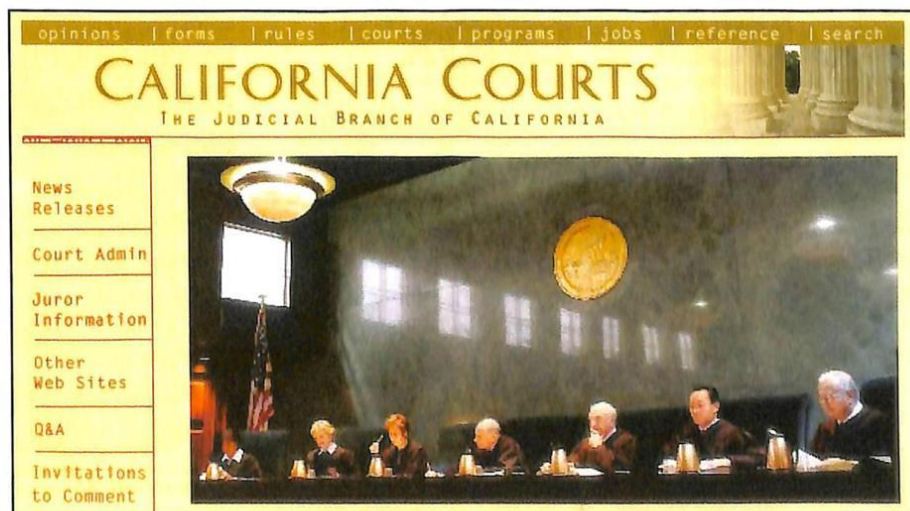
IBM rilascia come Open Source la piattaforma di sviluppo "Eclipse".

Dopo Netscape e Sun, anche IBM si unisce al gruppo di "benefattori" dell'open source, con un prodotto il cui costo di sviluppo è stato valutato in 40 milioni di dollari. Il progetto sarà gestito da un'organizzazione che includerà rappresentanti di diverse società di software. Eclipse fornisce una visione integrata del processo di sviluppo software e integra in un unico ambiente i diversi elementi necessari per la progettazione, lo sviluppo, la verifica, e il debug. La piattaforma è disponibile sia per Linux che per Windows, permettendo così per la prima volta ai programmatori Linux di sviluppare applicazioni facilmente portabili tra i due sistemi operativi. I primi strumenti a essere resi disponibili in Eclipse saranno quelli della famiglia IBM "WebSphere".

Per maggiori informazioni:
www.businesswire.com/cgi-bin/f_headline.cgi?bw.110501/213092380&ticker=IBM
www.eclipse.org

LIBERTÀ DI PAROLA, LIBERTÀ DI PROGRAMMAZIONE

Lo stato della California invoca le libertà costituzionali a tutela della pubblicazione in Internet del DeCSS.

"Pertanto, concludiamo che l'ingiunzione preliminare che proibiva all'imputato di pubblicare il DeCSS possa essere caratterizzata come una proibizione di pura espressione".
Con queste parole, la corte dello stato della California conclude che pubblicare un programma in formato sorgente su Internet è semplicemente "un'espressione scritta delle idee dell'autore", e come tale è tutelato dal Primo Emendamento della Costituzione. Sfortunatamente, il ragionamento non è applicabile nel caso di materiale coperto da diritto d'autore, e quindi neanche a leggi come la famigerata "Digital Millenium Copyright Act", utilizzata in altre denunce analoghe. Pure, vedere il diritto di espressione utilizzato a tutela del software libero non può che essere visto come un buon segno, soprattutto in un periodo in cui negli Stati Uniti sembra che non si faccia altro che promulgare leggi estremamente problematiche per il software libero, come la DMCA e la SSSCA.

Per maggiori informazioni:
www.courtinfo.ca.gov/opinions/documents/H021153.PDF

Accordo raggiunto tra Microsoft e Governo degli Stati Uniti

Si chiude con un accordo deludente la causa contro Microsoft per abuso di monopolio.

I singoli stati avranno probabilmente ancora da dire, ma a livello federale il processo Microsoft è concluso: complice anche la nuova amministrazione, verosimilmente meno accanita verso un monopolista, e in particolare contro questo , di quanto non potesse esserlo la precedente, il Dipartimento della Giustizia ha raggiunto un accordo con la società di Redmond. Alcuni punti di vantaggio per Linux e il software libero sembrano esserci. Il primo esempio è il divieto per Microsoft di penalizzare un OEM (Original Equipment Manufacturer, insomma un produttore di PC) che venda computer su cui sia installato un altro sistema operativo oltre Windows. D'ora in poi, quindi, dovrebbe essere più semplice trovare sul mercato i sistemi dual boot essenziali per un utente che voglia sperimantare il grande salto; resta da vedere, naturalmente, quali produttori saranno veramente interessati a questo difficile passo, visto il piccolo vantaggio che ne trarranno e la possibilità che Microsoft riesca comunque a trovare un modo di aggirare l'accordo. Secondo, gli OEM potranno sostituire elementi middleware prodotti da Microsoft con quelli di altri produttori; questo include prodotti come Internet Explorer, Outlook Express, e Java. Alcune delle clausole appaiono sospette, ma qualche piccolo progresso potrebbe esserci. Viene anche specificato il periodo entro il quale la Microsoft dovrà rendere disponibili le funzionalità per rimuovere gli elementi middleware dal sistema operativo, ma sempre con grandi margini di discrezionalità. Terzo, Microsoft dovrà pubblicare i prezzi delle licenze praticati agli OEM. "Pubblicare", non rendere pubblici: questo significa che soltanto gli OEM, anzi, soltanto i grandi OEM, avranno accesso a questi dati. Lo scopo di questa clausola è quello di impedire le alleanze che sono state strette in passato, o quanto meno renderle più difficili. Quarto, ma non per importanza: Microsoft dovrà rendere pubbliche le interfacce di programmazione tra il middleware e il sistema operativo, e dovrà rendere disponibili con una licenza "ragionevole e non discriminante" qualunque protocollo di comunicazione utilizzato per l'interoperabilità tra client e server in una rete Microsoft. Sembrerebbero grandi notizie per il progetto Samba, che si occupa proprio di queste problematiche, e che finora è stato costretto a non semplici contorsioni di reverse engineering; sfortunatamente, come i rappresentanti del progetto Samba si sono affrettati a indicare, esiste una clausola di esclusione, quella relativa ai "protocolli utilizzati per l'amministrazione remota", ed è la Microsoft stessa a decidere quali protocolli appartengano a questa categoria: una discrezionalità che certo non mancheranno di utilizzare.

Per maggiori informazioni:
www.theregister.co.uk/content/4/22684.html
news.cnet.com/news/0-1003-200-7763195.html
news.cnet.com/news/0-1003-200-7758623.html

MICROSOFT: LA MINACCIA È LINUX

"Linux è la minaccia a lungo termine contro il nostro core business. Non dimenticatelo mai!"

Un altro memo interno trapelato dalla Microsoft illustra, per chi si rifiutasse ancora di crederci, la preoccupazione del gigante dell'informatica nei confronti di Linux. "I nostri clienti stanno abbandonando le soluzioni basate su costosi UNIX proprietari" dice Brian Valentine, vice presidente della Divisione Windows", "sfortunatamente, siccome Linux è molto simile a UNIX [...] stiamo iniziando a vedere i clienti che spostano le loro applicazioni UNIX verso Linux su Intel". Valentine prosegue suggerendo di visitare i centri di calcolo dei clienti, individuando le soluzioni basate su UNIX; "tenete questo inventario nel portafoglio, tatuatevelo sul fondo schiena se necessario [...] sbatteteli fuori una macchina, un'applicazione, un dipartimento per volta".
Nel caso qualcuno dovesse dubitare della serietà con cui la Microsoft considera la "minaccia" Linux, il memo di Valentine può essere una lettura illuminante... "

Per maggiori informazioni:
www.theregister.co.uk/content/4/22770.html

AMAZON TAGLIA I COSTI GRAZIE A LINUX

Il gigante del commercio elettronico abbatte le spese informatiche da 71 a 54 milioni di dollari migrando i suoi server a Linux.

La migrazione è stata effettuata per contenere le perdite dei mesi passati. I server sono stati forniti dalla Hewlett-Packard, mentre il software e i servizi di installazione e gestione sono stati forniti dalla Red Hat, dominatrice indiscussa del mercato americano. Il caso di Amazon, quando saranno resi disponibili maggiori dettagli, può rappresentare un esempio interessante dei vantaggi economici derivanti dalla scelta di Linux. Rappresentanti della Microsoft hanno commentato che la scelta di Amazon comporterà a medio termine un incremento dei costi. Non si capisce bene a che titolo, dal momento che sembra la migrazione sia stata effettuata da server Unix piuttosto che da Windows NT.

Per maggiori informazioni:
news.cnet.com/news/0-1003-200-7720536.html

La supergriglia dei pinguini

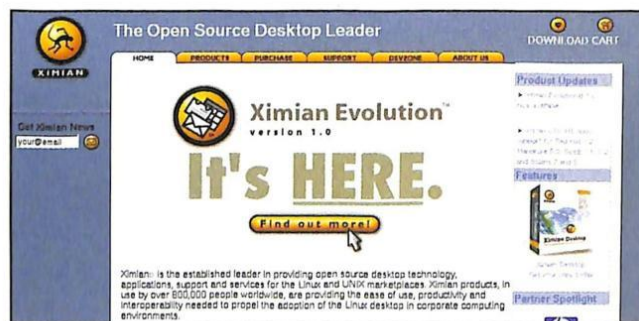
Sarà basato su Linux un superprogetto per l'analisi dati dei grandi esperimenti della fisica delle particelle.

Già da alcuni anni il CERN di Ginevra, il Laboratorio Europeo per la Fisica della Particelle nonché sede di sviluppo originaria del Web, ha iniziato una massiccia migrazione di buona parte dei sistemi di calcolo su Linux. Del resto, è ben noto che il pinguino trova buona accoglienza prima di tutto negli ambienti universitari e della ricerca. Nei prossimi anni gli esperimenti del "Large Hadron Collider" (LHC) genereranno una mole di dati impossibile da analizzare per un centro di calcolo tradizionale. Il laboratorio ha quindi avviato il progetto DataGRID, una "griglia" di computer basati su Linux distribuiti tra le centinaia di istituzioni partecipanti agli esperimenti in tutto il mondo. LHC genererà una mole di dati dell'ordine di alcuni PetaByte (milioni di GB) per anno. Al momento attuale, non esiste in Europa un sistema di calcolo in grado di analizzare una simile mole di dati. A differenza di sistemi come il SETI per la ricerca di intelligenze extraterrestri, il DataGRID effettuerà un'analisi on-line e in tempo reale. Si prevede che il progetto sia attivo per il 2006.

Per maggiori informazioni:
www.cern.ch
www.eu-datagrid.org

XIMIAN, STRATEGIA VINCENTE?

Non suscita grandi polemiche l'inserimento di un add-on proprietario nel software di comunicazione "Evolution".

La Ximian, distributrice di "Ximian Gnome" e "Ximian Red Carpet", ha rilasciato la versione 1.0 di Evolution. Il programma integra le funzioni di client di posta elettronica, gestione dei contatti e calendario: l'equivalente di Microsoft Outlook in Linux. Il prodotto ha ricevuto commenti estremamente positivi. Contemporaneamente, è stata annunciata la disponibilità, "all'inizio del 2002", di "Ximian Connector", un add-on che consentirà a Evolution di dialogare con un server Exchange, integrandosi così con l'ambiente di comunicazione Microsoft. La novità più grande, comunque, è che mentre Evolution resta un vero software libero, rilasciato con licenza GPL, Connector è un prodotto proprietario, con una licenza basata sulle postazioni di lavoro e un prezzo non trascurabile: 69 dollari per una licenza, 599 per dieci, 1499 per 25. Non ci sono stati forti critiche alla mossa di Ximian, almeno per ora (e nessuno ha pensato di chiedere un parere alla Free Software Foundation): segno forse del fatto che la comunità ha iniziato a rendersi conto che le compagnie open source

necessitano di nuove strategie per sopravvivere. La collocazione strategica del prodotto è comunque perfetta: la parte fondamentale del prodotto resta libera, e disponibile per l'utilizzo in un ambiente libero completo; le aggiunte proprietarie sono confinate alla sola integrazione con altri prodotti comunque proprietari, per società interessate a un ambiente misto. Per la sopravvivenza della Ximian resta da vedere quanto interesse vi sarà in soluzioni miste del tipo indicato, ma questo solo il mercato potrà dirlo.

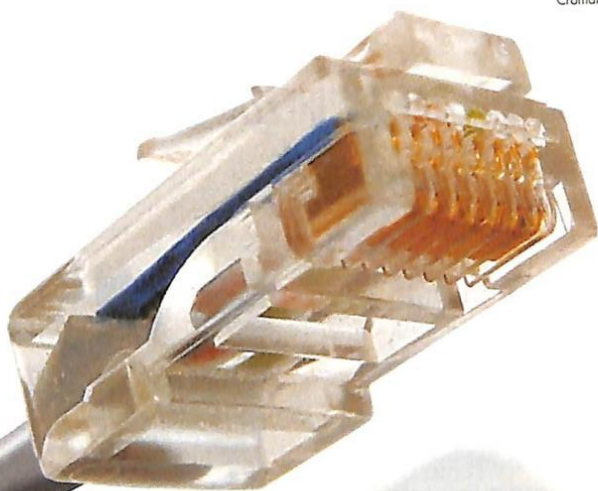
Per maggiori informazioni:

www.businesswire.com/cgi-bin/f_headline.cgi?bw.120301/213370452

www.businesswire.com/cgi-bin/f_headline.cgi?bw.120301/213370470

www.ximian.com

Cromatika

il magazine *con*nesso

GO! ONLINE

alla scoperta della rete

internet magazine

RIVISTA+CD+LIBRO
SOLO € 3,90

Febbraio 2002 • Periodicità Mensile

Siti in 3D
per un web sempre più reale

Intranet
Come usarle bene per trasformarle in uno strumento di lavoro utile

Motori di Ricerca
Quali sono i migliori e come si evolveranno?

Più comodi e convenienti che nella realtà, sono i

centri commerciali on line

Senza fili
Il Web è in cerca di **nuovi dispositivi...**

WEBMASTER
Guadagnare visibilità nei motori di ricerca grazie alle **pagine doorway**

Uso avanzato dei **layer** per migliorare le proprie pagine web

PRATICA
Installazione e configurazione di **IIS** per **Windows 2000**

Tracciare gli indirizzi e i nomi di dominio con **Mcafee Visualtrace**

Usare **Javascript** per adattare le pagine al browser e alla risoluzione

Cellulari che passione!
Dove trovare le dritte giuste per la scelta e per l'uso del proprio telefonino

Sei sempre in Rete, nel sito giusto
e al momento giusto, attraverso
tutorial, dossier, recensioni hardware
e software, articoli di tecnica.
Trovi sempre un nesso con i tuoi
interessi e le tue inclinazioni.
Allora, non aspettare, corri in edicola
e poi go online!

LINUX MASSIMA SICUREZZA

di anonimo
Edito da: Apogeo

Aspetti Generali

Un libro insostituibile per coloro devono utilizzare Linux in totale tranquillità, protetti tutti gli attacchi! Ormai è noto che Linux è il sistema migliore da utilizzare come server. Ma i servizi Web, lo scambio di file, la gestione di accessi telnet e i servizi di database mettono in primo piano il problema della sicurezza, aprendo porte e consentendo connessioni esterne.
Ad approfittare di questi falli potrebbero essere (anzi, sono) dei mailintenzionati il cui scopo è quello di accedere ai sistemi, visualizzare dati altrui e, nei casi peggiori, cancellarli provocando dann spesso irreparabili. "Linux Massima Sicurezza" è un testo che non deve assolutamente mancare nella libreria del bravo amministratore di rete. Nel libro è possibile trovare tutte le informazioni per aumentare la sicurezza del proprio sistema, imparando quali sono i "buchi" più conosciuti e quelli meno noti. Si apprenderanno le tecniche utilizzate dagli hacker per penetrare nei sistemi Linux e i metodi per proteggersi da questi attacchi.
Scritto proprio da un gruppo di persone esperte in tecniche di cracking.

Scheda

Titolo: LINUX MASSIMA SICUREZZA

Autore: Anonimo

ISBN: 88-7303-616-3

Prezzo: € 40,28

Livello: Avanzato

Edito da: Apogeo

Sito: www.apogeonline.com

Per Utenti Avanzati

Un testo adatto a sistemisti, amministratori di rete e utenti Linux che desiderano proteggere i loro server e le loro workstation dalle intrusioni indesiderate e da tutti quei problemi che minacciano l'integrità del sistema.

Gli Autori

Il testo è scritto da hacker esperti in tecniche di cracking che, conoscendo alla perfezione i punti deboli del sistema, mettono a disposizione la loro esperienza a risolvere questi problemi di vulnerabilità.

Aspetti Specifici

Tecniche avanzate di configurazione di Linux per eliminare i problemi di sicurezza; i buchi e le tecniche di cracking più diffuse; programmi e utility in grado di rafforzare la sicurezza; gli strumenti che utilizzano i cracker; migliorare la sicurezza con tecnologie come SSL e MD5. "Linux Massima Sicurezza" vi svelerà tutti i punti deboli del sistema.

ConclusionI

Non sapete cos'è lo "spoofing" o vi siete sempre chiesti cos'è uno "sniffer"?
Allora correte in libreria a comprare questo libro!

PROGRAMMARE CON PHP 4

di Thomas Theis
Edito da: Apogeo

Aspetti Generali

PHP è il linguaggio rivoluzionario per la generazione di pagine Web dinamiche, pagine che possono modificarsi a seguito di azioni intraprese dall'utente.

Aspetti Specifici

Le caratteristiche di PHP, un ottimo linguaggio per la gestione ottimale dei form e l'interazione con diversi sistemi di gestione di database, tra cui MySQL con cui si integra alla perfezione.

Per Utenti Avanzati

Un testo adatto ad una fascia di utenza Medio-Alta.
Questo libro si propone di introdurre al lettore tutte queste caratteristiche e spiegare le novità della versione 4.

Scheda

Titolo: PROGRAMMARE CON PHP 4

Autore: Thomas Theis

ISBN: 88-7303-778-X

Prezzo: € 21,69 (i.i.)

Pagine: 312

Livello: Medio-Alta

Edito da: Apogeo

Sito: www.apogeonline.com

Conclusioni

Gli esercizi a corredo permettono di consolidare le conoscenze a acquisite.

Il bello del Pinguino...

Carissimi lettori di LinuxMAGAZINE, siamo molto contenti dell'interesse, che state dimostrando per la rivista! Come sempre siamo contenti di darvi la possibilità di approfondire alcuni aspetti della "linux-way" alla filosofia di sistema e alle possibilità di crescita personale che vengono messe a disposizione di chi si incammina lungo la "via del pinguino...". Quindi continueremo a rispondere alle mail inviate all'indirizzo linuxmag@edmaster.it (sintetizzandole per avere più spazio per le risposte), cercando di approfondire gli argomenti o le curiosità di interesse più generale.
linuxmag@edmaster.it
Ovviamente, per motivi riguardanti la privacy, riporteremo nomi e indirizzi di e-mail dei lettori che ci hanno scritto esattamente come riportato in fondo alla lettera.

da: Massimo

Easy Linux 2.4

Nell'ultimo numero della vostra rivista (dicembre-gennaio n. 15) ho molto apprezzato il CD-ROM allegato con la distribuzione easyLinux 2.4, che mi sembra effettivamente molto facile da installare, e molto adatta alle mie esigenze. A questo punto avrei voluto acquistarla integralmente, ma l'URL di riferimento (www.easyLinux.com) non funziona da molti giorni. Potreste aiutarmi ad individuarla, se fosse cambiata, o suggerirmi qualche pista per comperare la distribuzione?
Grazie,

Massimo.

Prova a consultare il sito del produttore *www.elT.de.*
Qualche settimana fa abbiamo provato a contattarli ma per lungo tempo non hanno risposto. Successivamente si sono fatti sentire e ci hanno spiegato che a breve rilasceranno una nuova versione migliorata della loro distribuzione che noi non mancheremo di allegare non appena la metteranno a nostra disposizione.

da: Giampietro

Debian e non solo...

Gentile redazione sono un Vostro assiduo lettore che non riesce a capire come mai una rivista interessante e ben fatta sia bimestrale, quando, nel panorama editoriale si tende alla pubblicazione (giornaliera, o quasi) di ogni tipo di cose che preferisco non nominare? Bravi ben fatto, DEBIAN é una distribuzione veramente allettante per noi linusini, ma scusate, non vi sembra una crudeltà mentale pubblicare la guida all'installazione a puntate. Nella speranza di poter installare correttamente DEBIAN entro il 2002 porgo auguri di buon Natale e Felice anno nuovo.

Giampietro

Grazie Giampiero. Il 2002 porterà delle sorprese. E, siamo certi, sarai in grado di installare la Debian. Quanto alla crudeltà... hai perfettamente ragione. Ma d'altronde: Debian deve essere difficile! O no?

da: Iperbole

Debian solo 1° parte?

Prima di tutto i complimenti per questa ottima rivista (sfortunatamente ancora bimestrale). Finalmente ho trovato allegata a Linux Magazine di dicembre la distribuzione Debian Gnu/Linux (potato) di cui ho sentito parlare molto bene per le sue incredibili potenzialità, ma perché il CD della distro è disk 1? La distribuzione è stata divisa in due CD? Devo quindi aspettare febbraio 2002 per averla tutta intera? Vi prego illuminatemi.

Iperbole

Aspetteresti inutilmente. Abbiamo voluto pubblicare Debian per colmare una lacuna che ci portavamo dietro fin dalla nascita della rivista e che stava diventando un po' troppo vistosa, tanto che qualcuno congetturava che, essendo una pubblicazione commerciale, eravamo pagate delle oscure forze del mercato cattivo per boicottare la distribuzione autenticamente free. Ovviamente non è cosi ma, come tutti i lettori possono essersi adesso accorti Debian è ben lontana dall'essere una distribuzione desktop e consumer. Con il primo disco, però è possibile effettuare una installazione completa e successivamente utilizzare

da: ***Amedeo***

Problemi di font

Sull'ultimo numero di Linux Magazine (Dicembre 2001-Gennaio 2002) ho notato la lettera di Gaetano avente per oggetto: Problemi di font. Essendo io stesso incappato, quale utilizzatore di Mandrake 8.0, nello stesso problema e non avendo risolto nulla definendo l'anti aliasing dei caratteri sotto l'ambiente KDE, ho provato a modificare il file /etc/X11/XF86Config (/etc/X11/XF86Config-4 nel caso si utilizzi XFree86 4) sostituendo la riga:

```
FontPath "unix/:-1"
```

con le seguenti:

```
FontPath "/usr/X11R6/lib/X11/fonts/100dpi"
FontPath "/usr/X11R6/lib/X11/fonts/75dpi"
FontPath "/usr/X11R6/lib/X11/fonts/PEX"
FontPath "/usr/X11R6/lib/X11/fonts/Speedo"
FontPath "/usr/X11R6/lib/X11/fonts/Type1"
FontPath "/usr/X11R6/lib/X11/fonts/drakfont"
FontPath "/usr/X11R6/lib/X11/fonts/encodings"
FontPath "/usr/X11R6/lib/X11//X11/fonts/mdk"
FontPath "/usr/X11R6/lib/X11/fonts/misc"
FontPath "/usr/X11R6/lib/X11/fonts/mozilla-fonts"
FontPath "/usr/X11R6/lib/X11/fonts/pcf_drakfont"
```

risolvendo, almeno nel mio caso, il problema. Hope this helps! Cordiali saluti.

Amedeo

gli strumenti automatizzati per scaricare tutte le aggiunte e gli aggiornamenti direttamente da rete.

da: ***Emanuele***

Manuali Linux

Salve mi presento, mi chiamo Emanuele Moranti, ho sottomano la vostra rivista Linux Magazine, mi piace come é strutturata e l'idea di linux, mi piace tantissimo ma c'é un problema, a me mancano completamente le basi.
Potreste indicarmi il nome di un libro, un manuale in italiano che mi possa dare queste basi per cortesia? Vi ringrazio anticipatamente.

Emanuele

In libreria esistono ormai moltissimi libri per iniziare. Ad esempio "LINUX - Installazione, configurazione, Uso" di Michael Kofler della Addison Wesley Italia (50,61 EUR) oppure "Linux - Guida all'Utilizzo" di Guidi, Martiradonna e Vitiello della Hoepli (22,72 EUR). In rete poi ci sono molte guide libere che non hanno veramente nulla da invidiare ai libri in commercio. Per esempio Appunti Linux di Daniele Giacomini (*http://a2.swlibero.org*), oppure Linux Domande e Risposte di Gaetano Paolone o Linux Facile di Daniele Medri (*www.linuxfacile.org).*

[successivo] [precedente] [inizio] [fine] [indice generale] [violazione GPL] [licenze] [indice analitico]

Appunti di informatica libera

da: ***Antonio***

Complimenti Gianluca!

Caro Gianluca, ti scrivo a proposito dell'articolo "Programmi di masterizzazione per Linux", sul numero 14 della rivista. Complimenti. Poche volte ho letto articoli così chiari e immediatamente fruibili. Alla prima operazione (con *mkisofs* e *cdrecord,* sono un vecchio smanettone, e preferisco controllare il processo da console) ho fatto quello che per mesi (senza molto approfondire, dovendo badare a una famiglia) ho tentato, spulciando gli how-to, e non trovando quello che hai trovato tu, e che hai esposto così chiaramente. Deduco quindi che:

1) devo senz'altro migliorare i miei metodi di ricerca, magari non facendoli alle due di notte (più o meno l'unico periodo libero della giornata), e cercando più in profondità.

2) Linux Magazine è una rivista senz'altro da seguire anche in futuro.

Ciao. Alla prossima.

Antonio

Per contattarci:

e-mail:
linuxmag@edmaster.it
Fax: 09848319225
Posta:
Edizioni Master,
C.da Lecco, zona industriale - 87030
Castiglione Scalo (CS)

Il diavoletto e il pinguino

Una veloce panoramica sulle caratteristiche essenziali dei sistemi BSD liberi.

"A parte il nome figo, qualcuno può spiegarmi perché dovrei usare Linux anziché BSD?"
"No. Questo è tutto. Il nome figo, voglio dire. Abbiamo lavorato duramente per creare un nome che piacesse alla maggioranza delle persone, e ne è valsa la pena: migliaia di persone usano Linux solo per poter dire, "OS/2? Ah! Io uso Linux. Che nome figo". 386BSD ha fatto l'errore di mettere un mucchio di numeri e strane abbreviazioni nel nome, e questo spaventa un sacco di persone solo perché suona troppo tecnico". (Linus Torvalds, rispondendo a una domanda su Linux e BSD).

Potenza della legge

La nostra rivista, come accade a molti che si occupano del mondo del software libero, ha ricevuto sia complimenti che critiche per il fatto di dare spazio non solo agli aspetti puramente tecnici, ma anche a quelli legali. Per tutti coloro che ritengono inutile o poco interessante occuparsi di argomenti di questo tipo, può essere interessante illustrare il motivo di base per cui Linux è divenuto molto più popolare dei cugini BSD.
I sistemi BSD *("Berkeley Software Distribution")* sono nati nell'Università di California, a Berkeley. Quando i Bell Labs svilupparono il primo UNIX, la AT&T era un monopolio controllato, e non poteva commercializzare sistemi informatici; il progetto fu quindi liberamente ceduto alle università, e l'università di Berkeley ha effettuato i successivi sviluppi che ha portato a quelli che oggi sono i sistemi BSD.

Figura 1

Il demonietto senza nome (a volte detto "Beastie") che costituisce il logo di FreeBSD nel salvaschermo della console.

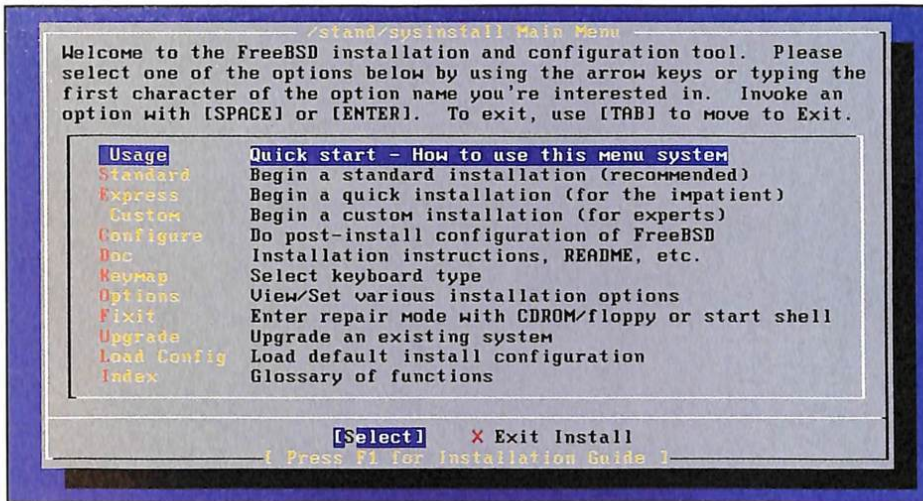



Figura 2

La schermata di benvenuto del programma di configurazione di FreeBSD.

Con la nascita del 386 all'inizio degli anni '90, il PC era pronto per un sistema operativo "serio". Nello stesso periodo in cui Linus Torvalds iniziava a scrivere il suo sistema operativo per hobby, la comunità BSD stava effettuando il "porting" del sistema per la nuova piattaforma. Sfortunatamente, nello stesso periodo la AT&T era stata ormai divisa e poteva tornare a occuparsi di informatica; per riguadagnare il controllo del mondo Unix, pensò bene di far causa all'Università della California. La causa giunse a un accordo dopo l'acquisto di UNIX dalla AT&T da parte della Novell, ma l'incertezza che era regnata nel settore fino a quel momento aveva fatto in modo che Linux guadagnasse una base di utenti molto più ampia di quella del progetto 386BSD.

"A differenza di Linux, nato nella cameretta di uno studente finlandese, FreeBSD è nato come progetto pubblico, il che ha portato a un forking molto prematuro"

Licenza e prestazioni

Se chiedete a uno sviluppatore BSD perché non usi Linux, le risposte probabili saranno "prestazioni" e "licenza". La comunità BSD, in linea di massima, odia la GPL ("General Public License"), la licenza scritta dalla Free Software Foundation e utilizzata in Linux oltre che in centinai di altre applicazioni libere. La differenza tra la GPL e la licenza BSD è fondamentale: la prima impone che qualunque prodotto derivato sia a sua volta rilasciato con licenza GPL, la seconda non dice assolutamente nulla dei lavori derivati. È questo che ha permesso alla Microsoft di "impossessarsi" di alcuni pezzi di codice BSD per l'inserimento in Windows; con Linux non avrebbero potuto farlo senza rischiare di essere portati in tribunale... e costretti a rendere libero l'intero codice di Windows!
In effetti, la licenza BSD può essere apprezzata dalla Microsoft in quanto monopolista, ma gli altri produttori di software possono essere meno entusiasti. La GPL fornisce una garanzia non solo verso le comunità di sviluppatori liberi, impedendo che il loro lavoro venga incorporato in prodotti commerciali inaccessibili, ma protegge anche le società di informatica le une dalle altre. Con la licenza BSD qualunque compagnia abbia un minimo vantaggio competitivo può provare a raggiungere posizioni di monopolio; la GPL ha invece una tendenza riequilibratrice, perché nessuno può effettuare modifiche senza renderle pubbliche, e quindi diventa impossibile sviluppare l'equivalente di Windows. È questo che compagnie come la Sun e la IBM amano della GPL. Per quanto riguarda le prestazioni, benchmark su benchmark sono stati accumulati in giro per il mondo; e in linea di massima è abbastanza ben noto che in alcune aree le prestazioni dei sistemi BSD sono migliori di quelle di Linux. In particolare, la gestione della memoria virtuale in Linux inizia solo ora ad avvicinarsi all'efficienza di BSD; e il sottosistema della rete è più efficiente e meglio strutturato nei sistemi BSD che in Linux. Alcuni sviluppatori Linux mettono in dubbio il valore di questi benchmark, ma di fatto nessuno si è mai sognato di dire che Linux è superiore ai sistemi BSD; e di fatto nelle applicazioni professionali si tende a preferire questi ultimi.

L'albero genealogico

Come si evince dalla citazione ripor-

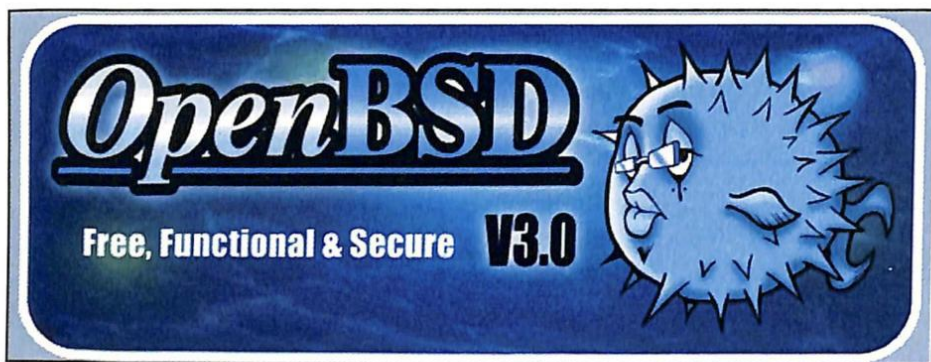



Figura 4

Il logo di OpenSSH.

tata all'inizio di questo articolo, il primo sistema BSD sviluppato per sistemi Intel aveva l'infelice nome di "386BSD". Il primo sviluppatore fu William Jolitz, all'inizio degli anni '90; il suo lavoro fu ripreso e portato avanti da Jordan K. Hubbard, Rod Grimes e Nate Williams, e finì per essere ribattezzato "FreeBSD" da David Greenman della Walnut Creek, produttrice di CD contenenti software libero. La prima versione ufficiale di FreeBSD fu la versione 1.0 nel dicembre del 1993; a novembre del 1994 usciva la versione 2.0, libero dalle sezioni di codice implicate nella citazione della AT&T.

A differenza di Linux, nato nella cameretta di uno studente finlandese, FreeBSD è nato come progetto pubblico, il che ha portato a un forking molto prematuro. Le persone insoddisfatte del lavoro svolto su FreeBSD hanno creato un progetto parallelo, denominato "NetBSD", rapidamente focalizzatosi sul porting a piattaforme diverse da Intel. Al momento attuale, NetBSD gira su 46 diverse piattaforme, dall'Atari alla PlayStation 2, dal PC all'alpha 64, ed ha costituito la fondazione per il lavoro di porting di numerosi altri progetti, non escluso Linux.

La terza piattaforma, OpenBSD, nasce da un disaccordo fra *Theo de Raadt*, responsabile della piattaforma SPARC, e il resto del gruppo di sviluppo NetBSD. La divisione avvenne intorno alla fine del '95, e la prima versione di OpenBSD fu rilasciata nell'ottobre del '96. Essendo residente in Canada, *de Raadt* non era limitato dalle leggi statunitensi sull'esportazione di codice crittografico, e ne ha approfittato per focalizzare lo sviluppo di OpenBSD sulla sicurezza. In effetti, OpenBSD utilizza i più avanzati algoritmi di crittografia; ha dato vita allo sviluppo di OpenSSH, l'implementazione libera del celebre protocollo SSH; e ha portato al mondo del software libero la pratica dell'auditing del codice, il controllo linea per linea volto a identificare ed eliminare i problemi di sicurezza.

Un'altra conseguenza della storia di BSD rispetto a Linux è il diverso modello di sviluppo; in qualità di sviluppatore originario, Linus Torvalds si trova ad essere il "dittatore benevolo" del kernel, la persona che prende le decisioni finali su cosa inserire nel kernel. Nei sistemi BSD, invece, le decisioni sono maggiormente decentralizzate, con un'ampia base di sviluppatori, un nucleo più ristretto ma comunque non piccolo di committers, autorizzati ad aggiornare i sorgenti del progetto, una squadra di nucleo. I sorgenti sono mantenuti su un server CVS, aggiornato svariate decine di volte al giorno, che consente l'accesso in maniera molto semplice a qualunque versione del sistema.

Un'importante differenza tra Linux e i sistemi BSD è che Linux di per sé costituisce solo il kernel, mentre i progetti BSD sviluppano anche un certo numero di

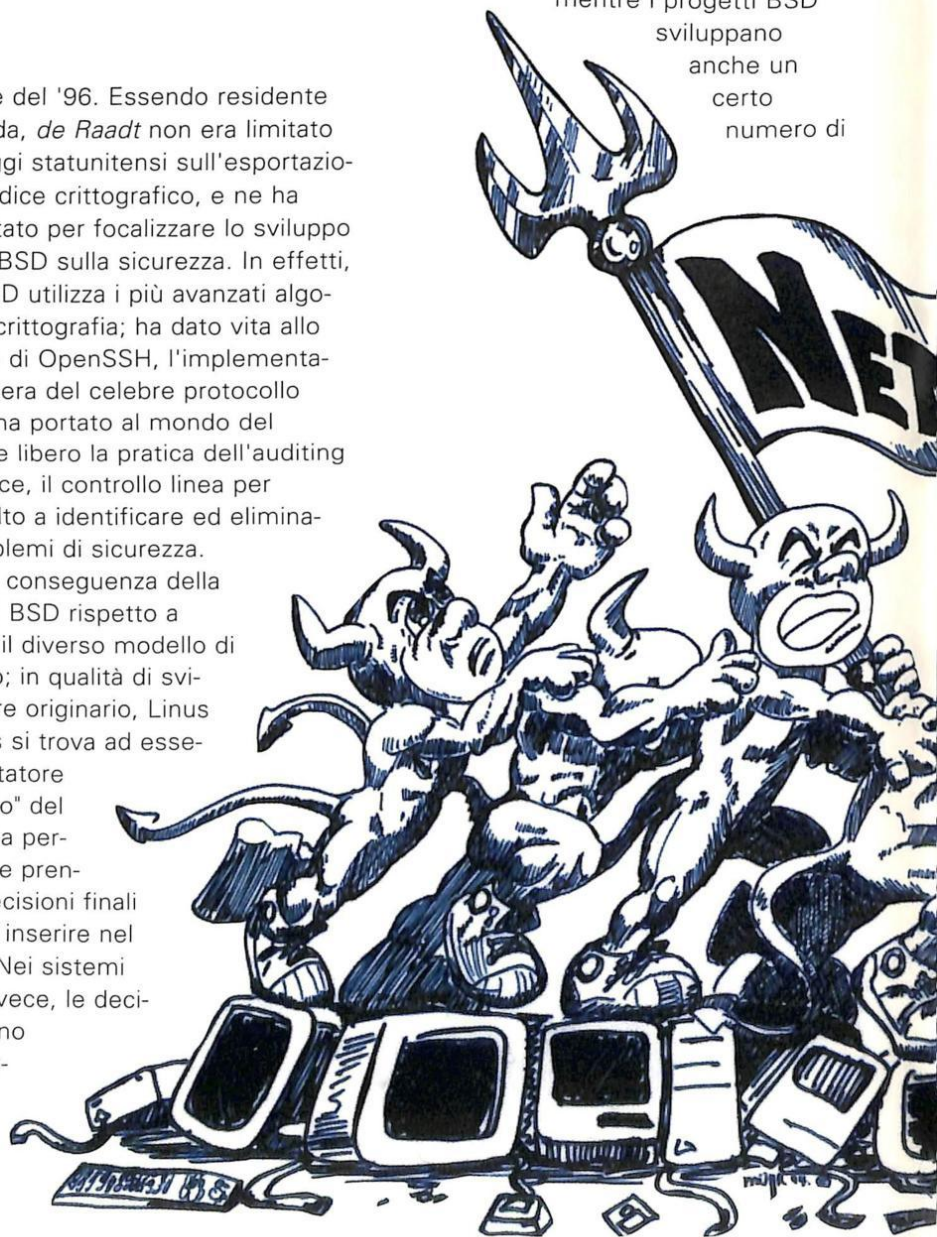



Figura 3

Il logo di NetBSD riprende il tema dei demonietti e lo combina con l'idea di uno sforzo di squadra su molteplici computer, magari anche vecchi.

Figura 5

FreeBSD è normalmente installato in un'unica partizione, all'interno della quale vengono ritagliate delle fette ("slice") dove vengono creati i diversi filesystem.

strumenti di base. I progetti BSD sono quindi completati attraverso il meccanismo dei port, centinaia di applicazioni libere compilate per BSD al di fuori del gruppo di sviluppo; mentre Linux, com'è noto, è completo grazie al lavoro dei distributori.

Linux contro FreeBSD

Recentemente ho avuto occasione di utilizzare FreeBSD per un impiego professionale, e sono rimasto estremamente impressionato dalla semplicità di installazione, configurazione e gestione. La semplicità di installazione non è paragonabile a distribuzioni Linux come Mandrake o easyLinux, ma non crea problemi a un utente Linux con un minimo di esperienza: non ci sono difficoltà particolari, è sufficiente avere una certa familiarità con la terminologia generica dei sistemi unixoidi. E soprattutto, quel che manca in semplicità viene recuperato in flessibilità.
La disposizione dei file di configurazione è estremamente semplice e lineare, e per molte configurazioni esiste comunque uno strumento automatizzato, raggiungibile attraverso il programma di configurazione */stand/sysinstall*: lo stesso programma, tra l'altro, utilizzato per la configurazione in fase di installazione, caratteristica banale ma che poche distribuzioni Linux condividono.
Il sistema di gestione dei pacchetti è uno dei punti di forza di FreeBSD. Sotto questo aspetto ricorda la Debian, al momento attuale la mia distribuzione preferita, ma con un'interfaccia a linea di comando decisamente più completa e semplice.
Infine, è doveroso sottolineare un aspetto importante, anche se per molti lettori sarà scontato: le applicazioni nei sistemi BSD sono in linea di massima le stesse che in Linux. Il sistema grafico è sempre X11, tutti i window manager funzionano nei sistemi BSD come in Linux, e persino le applicazioni disponibili per Linux solo in formato binario, come ad esempio StarOffice, Oracle, VMWare, Opera, eccetera - girano anche in FreeBSD, non solo senza problemi, ma addirittura, sostiene la comunità BSD, con prestazioni migliori, grazie alla migliore architettura del kernel e al fatto che non è effettuata tramite emulazione.

Quale sistema?

Stante la situazione, perché non convertirsi a FreeBSD? Beh, innanzitutto bisogna chiedersi "perché sì"! Forse FreeBSD ha

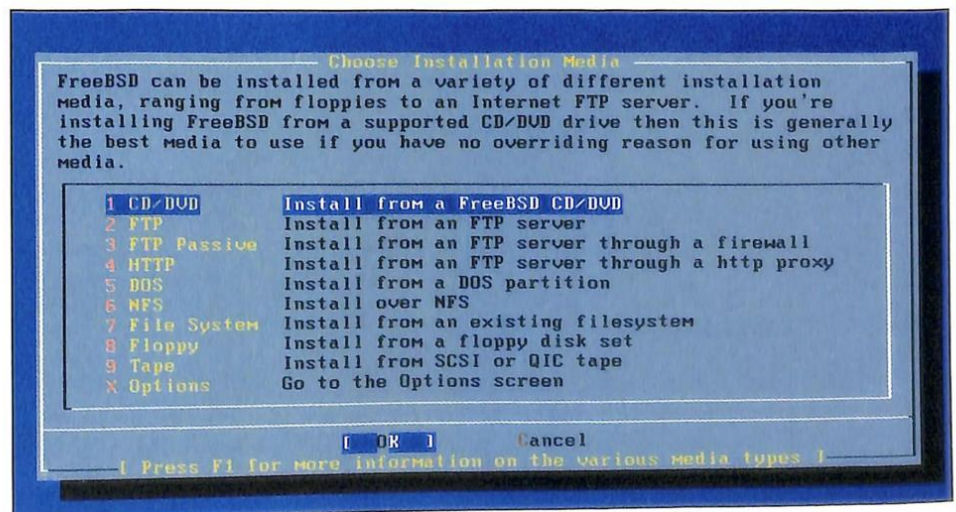



Figura 6

I diversi media da cui è opssibile installare FreeBSD.

prestazioni migliori di Linux, ma nella maggior parte dei casi la differenza sarà impercettibile. Se state già utilizzando un sistema libero, qualunque esso sia, non c'è motivo di cambiare, a meno che non abbiate forti preferenze ideologiche per una licenza o per un'altra, per un modello di sviluppo piuttosto che per l'altro; ma questa può essere una motivazione valida per uno sviluppatore o per una società di software, piuttosto che per un utente.

Se invece non state ancora utilizzando un sistema libero, bene, ogni momento è buono per iniziare, e ogni sistema libero va bene. Per utenti alle prime armi, comunque, è consigliabile una distribuzione Linux semplice e diffusa (in genere io suggerisco la Mandrake) piuttosto che un sistema BSD, sul quale è abbastanza difficile trovare aiuti esperti.

La situazione cambia per utilizzi professionali, in particolare per server di rete. In questo caso, se si ha esperienza sufficiente, un sistema FreeBSD può fornire prestazioni migliori di un sistema Linux equivalente.

Se ci si sente un po' avventurosi e si desiderano maggiori caratteristiche di sicurezza, vale la pena di prendere in considerazione OpenBSD. La scelta di NetBSD è probabilmente un po' più esotica, ma può essere la strada migliore se si ha necessità di utilizzare un sistema con architettura non Intel.

Conclusioni

Dal punto di vista dell'utente finale, non ci sono grandi differenze tra Linux e i sistemi BSD, soprattutto con il più diffuso FreeBSD; in pratica, la differenza più grande è forse che è più facile trovare qualche amico esperto di Linux e disposto ad aiutarci, piuttosto che persone competenti di BSD. Un utente veramente esperto, del resto, non avrà serie difficoltà a passare a un sistema BSD. Dal punto di vista del programmatore, le differenze più importanti sono quelle relative alla licenza e al modello di sviluppo: la scelta della prima ha soprattutto una componente ideologica; per quanto riguarda la seconda, BSD può forse dare a lungo termine maggiori soddisfazioni, ma Linux ha sicuramente una maggiore visibilità esterna alla comunità. Dal punto di vista delle società di software, infine, la licenza GPL offre sicuramente garanzie più ampie per il mercato di quanto non faccia la licenza BSD, anche se questa può essere più gradita a società di sviluppo che desiderino copiare codice senza dare nulla in cambio.

In ogni caso, l'esistenza dei sistemi BSD, e in particolare di FreeBSD, è un pensiero consolante nei momenti in cui si ha il timore che lo sviluppo di Linux imbocchi un vicolo cieco, venga completamente fuorviato, oppure prenda per qualche motivo una piega proprietaria. Un informatico evidentemente non ben consapevole dei meccanismi del software libero mi diceva, qualche mese fa: "Ah! Hai visto? Linux è stato acquistato dall'IBM! Sapevo che sarebbe finita così!".

Non riuscendo a convincerlo che le cose non stavano precisamente in questo modo, sono riuscito a zittirlo solo concludendo "Vabbè, vuol dire che inizierò a usare FreeBSD..."

Francesco Marchetti-Stasi

"Una importante differenza tra Linux e i sistemi BSD è che Linux di per sé costituisce solo il kernel, mentre i progetti BSD sviluppano anche un certo numero di strumenti di base"

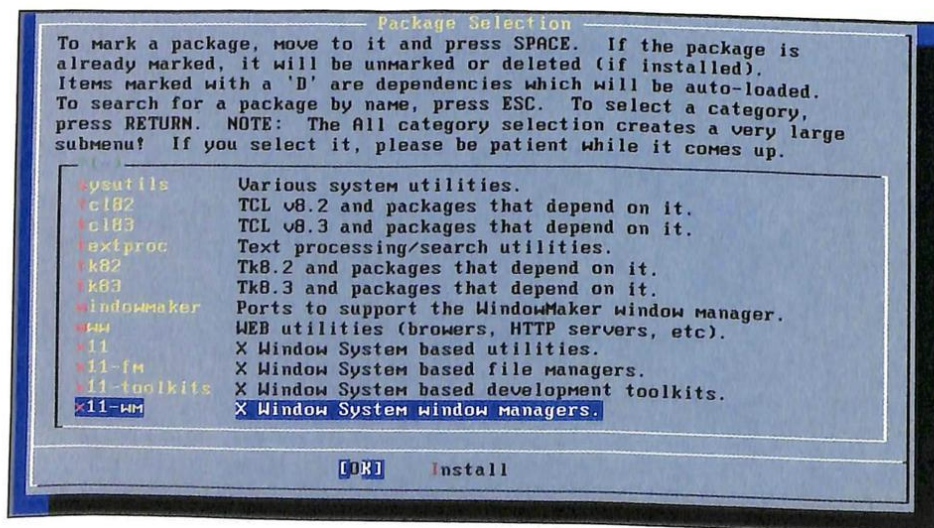
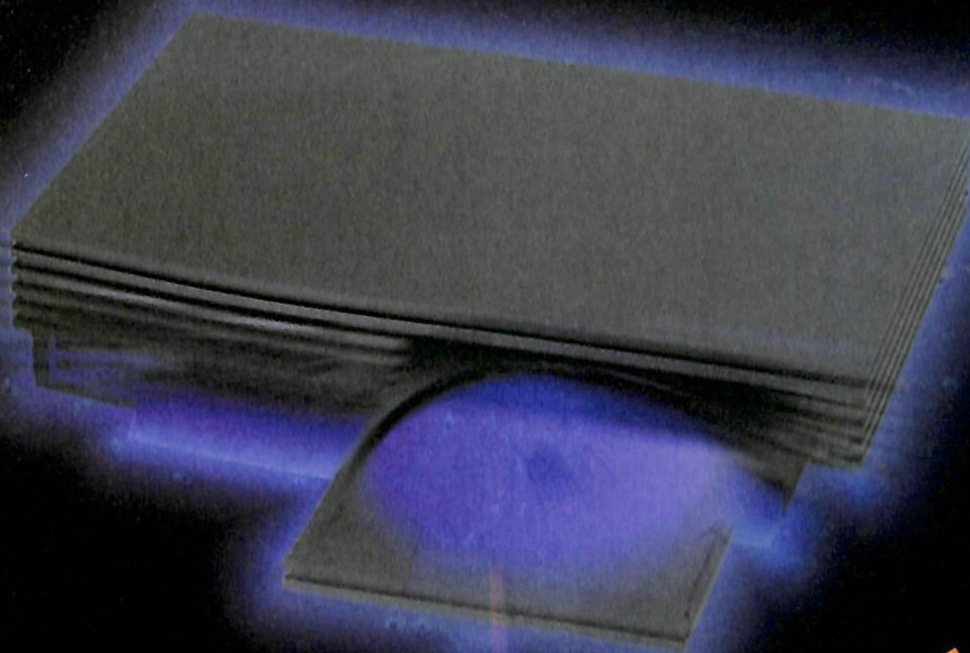



Figura 7

Il sistema di gestione dei pacchetti di FreeBSD, nella modalità semigrafica utilizzata durante l'installazione.

L'evoluzione della specie

I "cugini" di Linux: gli Unix commerciali.

Linux è un sistema operativo dalle mille sfaccettature, ed una di queste consiste nel rappresentare una delle molte versione di Unix disponibili. Diamo un'occhiata a quelle attualmente in commercio.

Un'origine comune

L'albero genealogico di Unix ricorda quello tipico dell'evoluzione di una specie animale: si parte da un antenato comune, da cui si differenziano via via altre specie, alcune delle quali arrivano inalterate ai giorni nostri, altre si estinguono, altre ancora danno origine a nuovi rami evolutivi. Questo è esattamente quello che è successo con Unix: nato alla fine degli anni '60 nei laboratori della Bell, principalmente a opera di Ken Thompson e Dennis Ritchie, viene nei primi anni relegato all'interno dei laboratori della AT&T e nelle centrali telefoniche, dato che quest'ultima deteneva il monopolio delle telecomunicazioni negli USA e la legge federale gli impediva di invadere altre aree di mercato. Oltre ad utilizzare Unix internamente, AT&T decise di concederlo gratuitamente, con un tipo di licenza che oggi chiameremmo opensource, alle istituzioni universitarie e di ricerca. L'occasione si concretizzò quando Ken Thomspon prese un anno sabbatico per insegnare all'università di Berkley. Grazie alle sue qualità, alla estrema portabilità (essendo scritto in un linguaggio di alto livello quale il C), e alla possibilità di sbirciare nei sorgenti e, quindi, di aggiungere componenti a piacere, si diffuse rapidamente tra studenti e professori, dando origine alla versione BSD (Berkley Software Development) di Unix. Tutto questo durò fino a quando l'antitrust impose alla AT&T di dividersi in un certo numero di aziende indipententi (le Baby Bells), liberandole nel contempo dal vincolo di fare affari al di fuori del mondo telecom. Iniziò quindi a vendere Unix come prodotto, imponendo alle università una scadenza oltre la quale avrebbero dovuto smettere di usare Unix derivante da sorgenti AT&T. Questo spinse il mondo accademico a riscrivere praticamente tutti i componenti che nel frattempo avevano conservato pezzi di codice originale, dando origine ai progetti di FreeBSD, NetBSD, OpenBSD e alle versioni commerciali quali IBSD e BSD/Os. All'inizio degli anni '80 anche i principali produttori di hardware e software cominciarono a produrre e commercializzare le loro versioni di Unix, attratti da un mercato che si andava espandendo a spese di quello dei mainframes e dei mini; in special modo i produttori di hardware commercializzarono versioni di Unix ottimizzate per le proprie architetture. Alcuni erano derivazioni di BSD, altri si ispiravano a quello di AT&T, in particolar modo alla versione denominata System V.

La santa alleanza

Uno di questi produttori, Sun

Microsystems, si alleò (1987) con AT&T per produrre La versione definitiva di Unix, prendendo sia da System V, che da Sun Os, la più diffusa versione di BSD in circolazione. Gli altri produttori, allarmati da ciò, l'anno successivo si riunirono in un'associazione chiamata *Open System Foundation* (OSF), producendo dopo poco tempo OSF/1. Guerre commerciali a parte, l'esigenza di standardizzare in qualche modo i diversi dialetti del sistema operativo si era già concretizzata con la nascita delle specifiche POSIX, promossa da IEEE, e con il consorzio X/Open, da parte di un certo numero di aziende europee.

come *"BSD-oriented"* piuttosto che *"SystemV-oriented"*. Un utilizzatore di sistema di derivazione BSD, ad esempio, utilizzava la C-shell come default, chiamava *vmunix* il kernel, utilizzava i files in */etc* chiamati *rc.** per lo start dei servizi, stampava utilizzando i comandi *lpr, lpq, lprm* e controllava i processi con *ps –aux*. Un utente *System V*, invece, usava *ksh* come shell, chiamava unix il kernel, controllava lo stato dei processi con *ps –ef*, posizionava gli script di startup e shutdown nelle directory */etc/rc.d*, controllava la stampante con *lp, lpstat, cancel* ed era in grado di esaminare le caratteristiche del sistema curiosando nella directory */proc*. Gli Unix

Figura 1

Il logo di Solaris.

all'introduzione del supporto multiprocessore ha elevato il suo target di azione. Attualmente è possibile trovare macchine con 106 processori (!) e sistema operativo Solaris. È la versione commerciale di Unix attualmente più diffusa.

"Sun Microsystems, si alleò (1987) con AT&T per produrre La versione definitiva di Unix, prendendo sia da System V, che da Sun Os, la più diffusa versione di BSD in circolazione"

La contrapposizione tra i due schieramenti durò fino al 1993, quando AT&T abbandonò il computer business cedendo i diritti del marchio Unix a Novell, che a sua volta li cedette a X/Open. Nel 1996, infine, X/Open e OSF si fusero nell'Open Group, al quale aderì anche Sun, e che tutt'oggi promuove la standardizzazione degli Unix esistenti.

Sapori differenti

Nelle innumerevoli versioni esistenti di Unix si possono riconoscere le due diverse "anime": *System V* e *BSD*. Fino a qualche anno fa le differenze tra le due versioni influenzavano pesantemente i vari dialetti, e non era troppo difficile classificare ognuno

attualmente in commercio sono ormai ad un livello di evoluzione abbastanza elevato, ed incorporano perciò il meglio preso dai due diversi capostipiti. Si può notare, tra l'altro, che anche Linux eredita caratteristiche da entrambi i sistemi.

Solaris (Sun)

La sua versione corrente, la 2, al contrario dei sui predecessori, Sun Os e Solaris 1, si ispira principalmente a *System V Rel. 4*. Viene prodotto per architetture PC-Intel e compatibili, e soprattutto per Sun-Sparc. È in grado di integrarsi in maniera perfettamente trasparente in una rete Windows. Nato inizialmente come motore di workstation a basso costo, grazie

Hp-Ux (Hewlett Packard)

Ha esordito su macchine della serie 9000/500 con processore HP Focus, per poi spostarsi su HP 9000/300 con processori Motorola. Passando per la serie Apollo, si sposta definitivamente

Figura 4

Un server HP Superdome.

su processori PA-Risc, che rappresentano anche l'offerta attuale. È stato tra i primi ad offrire un'interfaccia grafica evoluta, il VUE,

oggi completamente rimpiazzato dal CDE, offerto anche da tutti gli altri concorrenti come standard desktop. Ha, inoltre, un buon programma di amministrazione chiamato *SAM (System Administrator Manager)*, che consente di eseguire alcune delle operazioni più frequenti in maniera estremamente semplice. Ha una forte impronta *System V*.

Figura 2

La famiglia degli Alphaserver Compaq.

Aix (IBM)

Nonostante la convivenza con diversi altri sistemi operativi in casa IBM, ha avuto una discreta caratterizzazione, con la conseguenza però di risultare meno "standard" di altri suoi concorrenti. D'altra parte, l'amministratore di sistema viene coccolato da un favoloso tool di nome *Smit*, che non solo consente di eseguire praticamente tutte le operazioni necessarie guidato da semplici menù, ma è anche in grado i vedere, in un'altra finestra, i comandi testuali che il sistema deve utilizzare per portarle a termine. La sua ultima versione è denominata *Aix 5L*, dove la L sta per ... Linux! Infatti la sua caratteristica principale consiste in una spiccata (e sbandierata) affinità col sistema del pinguino, che consiste in una compatibilità a livello sorgente, ed in una serie di strumenti atti a facilitare la migrazione di applicazioni Linux in ambiente Aix.

Figura 3

Aix 5L.

Tru64 (Compaq)

Diretto erede del Digital Unix, ha cambiato nome e brand quando Compaq ha acquistato Digital, assorbendone tutte le risorse e tecnologie, ma cancellandone il nome, e, con esso, un importante pezzo della storia dell'informatica. Tra i tanti meriti di *DEC (Digital Equipment Corporation)*, non dimentichiamo l'aver dato i natali a Unix, grazie al *PDP-7* dei Bell Labs dove lavorarono Thompson e Ritchie.
In ogni caso, il nome *Tru64* sta ad indicare la caratteristica principale di questo sistema, vale a dire una spiccata capacità di utilizzare l'architettura a 64 bit reali dei processori Alpha (true 64, 64 bit reali). Recentemente Compaq è stata acquistata da Hewlett Packard: che subisca la stessa sorte di Digital?

Tutti gli altri

Le versioni appena analizzate rappresentano una buona fetta del mercato, ma non bisogna dimenticare l'esistenza di un buon numero di concorrenti. Tra questi possiamo citare Irix di Silicon Graphics, Mac Os X di Apple, Darwin, Dynix/PTX.

Conclusioni

Il sistema operativo Unix ha rappresentato una vera rivoluzione nel mondo informatico, spazzando via un mondo basato su costosi sistemi proprietari e regalando agli utenti il concetto di "sistema aperto", in grado di interagire e di scambiare software con sistemi anche di diversi produttori. Inoltre non va dimenticato che Internet è nata ed esiste proprio grazie all'utilizzo di server Unix, e che il protocollo di rete TCP/IP è stato incorporato per primo in sistemi di questo tipo, e tuttora riescono a integrarlo meglio di chiunque altro. Molto probabilmente è proprio per questa sua intrinseca apertura che Stallman lo ha scelto come modello per il suo sistema GNU/Linux.
L'avvento di Linux ha rappresentato, allo stesso tempo, la consacrazione e la probabile condanna di Unix, almeno per come è stato inteso finora. Linux ha riassunto e mostrato tutte le caratteristiche migliori di Unix e, specie con l'ultima versione del kernel, ha raggiunto una maturità tale da raggiungere e spesso superare le versioni commerciali.
Gli unici vantaggi che una versione commerciale può oggi offire sono, essenzialmente, una struttura di supporto collaudata ed efficiente, e l'integrazione, la compatibilità e l'efficienza quando utilizzato con hardware RISC dello stesso produttore. Linux potrebbe rappresentare quello Unix definitivo che i diversi consorzi di produttori ambivano a produrre; IBM sembra aver mosso un importante passo in questa direzione, rendendo il suo Aix quanto mai vicino e intercambiabile con Linux, ma anche tutti gli altri produttori hanno mostrato un vivo interesse verso il pinguino. Una cosa è certa: Unix sarà sulla scena ancora per parecchio tempo.

Marcello Penna

Alla scoperta di Freenet

Che cosa è, come funziona e le procedure di installazione della nuova rete peer-to-peer

Dopo la breve premessa su questo nuovo sistema di comunicazione (Linux Magazine n.13), riprendiamo il discorso su Freenet, entrando più in dettaglio per quanto riguarda il suo funzionamento e le differenze con il web. Vengono inoltre riportate le informazioni necessarie per configurare la vostra Linux box come client e server Freenet, con requisiti di sistema e informazioni su come e dove installare, qualora la vostra postazione ne sia sprovvista, la Java Virtual Machine, necessaria per il funzionamento di Freenet. E' descritto qualche esempio di navigazione tra i documenti e per ultimo dove reperire informazioni sul web, e, perché no, come contribuire allo sviluppo di Freenet.

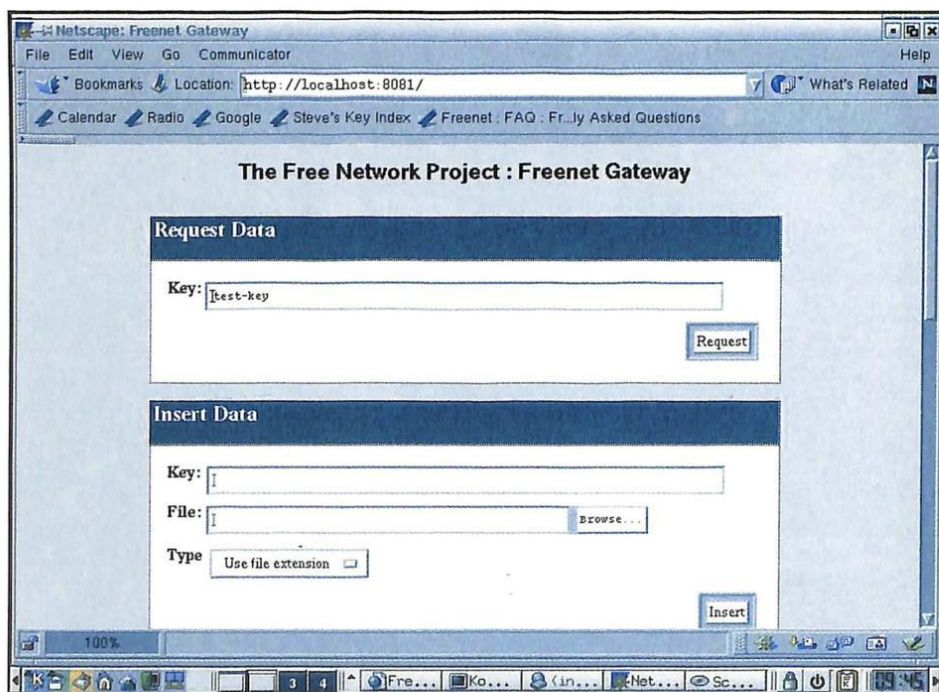



Figura 1
Il gateway di freenet con l'inserimento della chiave di prova test-key.

Evoluzione da rete centralizzata a rete decentrata

Come già spiegato nell'articolo precedente, Freenet si differenzia rispetto al web in quanto mentre in quest'ultimo l'informazione che consultiamo è disponibile presso un server specifico, in Freenet il documento ricercato è delocalizzato su un numero variabile di nodi. E' il concetto stesso di server che in questa nuova configurazione viene a cadere: ogni nodo può essere ugualmente client (ricerca e archivia documenti da altri nodi), e server (fornisce documenti su richiesta dei nodi vicini). Il vantaggio di questa configurazione paritaria fra nodi (in termine tecnico peer to peer), rispetto alla 'classica' rete web, sta nella disponibilità del documento: un server HTTP può essere messo fuori uso per problemi tecnici, infettato da virus che ne pregiudicano il funzionamento (vedi i famosi virus Code Red e Nimda), sottoposto ad attacchi tipo denial of service, o più semplicemente oscurato. Qualunque sia la ragione i dati in esso contenuti non sono reperibili in quei frangenti. Esistono due noti precursori delle reti peer-to-peer: Napster e Gnutella. Nel primo le informazioni di collegamento tra i differenti nodi è comunque centralizzato presso un server centrale, che potrebbe comunque essere soggetto ad attacchi di natura informatica o, come è noto, tentativi di oscuramento scaturiti dalla battaglia legale tra l'associazione delle compagnie discografiche americane e la società che gestiva Napster. Anche l'ultimo elemento di centralizzazione (l'informazione sulla reperibilità del file) viene rimosso con la rete Gnutella un progetto open source, dove ogni nodo (il vostro PC) è parte costituente di una rete a cui può appartenere come foglia terminale o come sottoramo da cui si dipartono nodi ulteriori. Questo vale ovviamente solo per la ricerca del file da scaricare perché le operazioni di download si svolgono tramite connessione diretta. Con freenet si va oltre: vengono aggiunte due novità che lo rendono estremamente interessante e veramente ....free. La prima è l'anonimato di chi richiede e/o fornisce informazione, la seconda è che capire dove risiedano le informazioni è molto difficile. Lo vediamo subito entrando nei dettagli.

Come funziona

Di seguito vengono illustrate le principali caratteristiche tecniche del funzionamento di freenet: dialogo fra i nodi, archiviazione ed estrazione dei documenti, controllo di autenticità dei documenti. Per maggiori dettagli tecnici sono disponibili i manuali on-line del codice e del protocollo di comunicazione su *http://freenet.sourceforge.net*. Come è già stato detto, la rete freenet non ha una rete gerarchica predefinita: ogni nodo fa sia da client che da server; inoltre ogni nodo conosce solo i propri vicini, che a loro volta sono collegati ad altri punti. I messaggi di trasmissione da nodo a nodo si compongono delle seguenti parti: nome, intestazione e dati. In quest'ultima parte è localizzato il contenuto del file. Questi messaggi viaggiano da un nodo all'altro, però, e questo è importante per la garanzia dell'anonimato, un nodo ricevente che abbia avuto una chiamata non può stabilire se il chiamante è effettivamente l'autore della richiesta oppure se fa a sua volta da intermediario. Bisogna però sottolineare che la garanzia di anonimato non è assoluta: una volta che il nostro nodo ha raggiunto il più vicino, comunica (solo a questo e non ad eventuali altri nodi che il vicino può chiamare), la propria identità. Resta comunque il fatto che non si può capire se la chiamata nasca dal nostro nodo o fungiamo da tramite per qualcun altro che rimane, questo si, sconosciuto. Una reale garanzia di anonimato si può ottenere solo cifrando opportunamente il pacchetto di dati sulla rete, come fanno gli anonymous remailer.

La struttura dei documenti: chiavi e hash

Ci sono una chiave ed un hash associati ad ogni documento presente in Freenet. Essi hanno il duplice scopo di garantire la ricerca e l'originalità del documento. Le funzioni hash generano, per un generico file in input un codice (targa) di lunghezza fissa. L'algoritmo che genera il file è non invertibile (dalla targa non si può ottenere il file di partenza), consistente (applicato ogni volta allo stesso file produce sempre la stessa targa), e *'collision free'* che vuol dire che due file differenti producono sempre delle targhe diverse. L'algoritmo utilizzato per freenet è il *SHA-1*. Esistono tre diversi tipi di chiavi in freenet: *CHK, KSK, SSK*.
CHK consiste in un hash generato dal documento con una chiave di cifratura generata in maniera casuale. Il documento così ottenuto è pubblicato con una chiave freenet composta dall'hash e dalla chiave di decodificazione. Ad esempio se si inserisce un documento con la forma seguente:

```
freenet:CHK@il_mio_file.txt
```

la chiave generata e da inserire per le successive ricerche sarà del tipo:

```
CHK@sdaTGu|uriTf..
```

Per verificare l'autenticità di un documento richiesto tramite questa chiave, ad ogni passaggio di nodo il file viene controllato decifrandolo con la sua chiave e viene riprodotto e confrontato l'hash ottenuto con quello della chiave. Per gli altri due tipi di chiavi freenet viene utilizzata la crittografia a chiave pubblica e privata. Tramite questo sistema di cifratura è possibile 'firmare' un documento in modo da garantirne con assoluta certezza la paternità: ogni coppia di chiavi è correlata, la chiave privata è segreta e serve per cifrare il documento. Il documento così modificato può essere decifrato esclusivamente tramite la chiave pubblica, inoltre non è tecnicamente possibile risalire alla chiave privata dalla sua 'sorella' pubblica. In questo modo cifrando il documento con la propria chiave privata, si dà la garanzia della sua provenienza, come se fosse appunto una firma, a chiunque sia in possesso della relativa chiave pubblica. Terminata questa breve digressione, comunque necessaria, vediamo come si applica questo a Freenet. Un esempio tipico di chiave *KSK* è il seguente:

```
freenet:KSK@freenet_come_funziona.txt
```

Il tipo di chiave è *KSK (Keyword Signed Key)* che utilizza il contenuto della chiave *(freenet_come_funziona.txt)* per generare una coppia di chiavi pubblica/privata secon-

do l'algoritmo *DSA;* il passo successivo è quello di produrre un hash del file. Invece la chiave privata viene utilizzata per 'firmare' il file. La chiave *SSK (SVK Subspace Key)* produce una coppia di chiavi *pubblico - privato* non sul singolo file ma su un namespace personale.
Ad esempio avranno una stessa coppia di chiavi i seguenti file:

```
freenet:SSK@linuxmagazine/
                    articoli/news.rtf
freenet:SSK@linuxmagazine/
                    articoli/freenet.rtf
```

Come freenet gestisce i documenti

Una caratteristica peculiare di freenet e che si tratta di una rete 'ecologica'. Proprio così i file vengono introdotti senza la possibilità da venire rimossi da nessuno. Ciascun nodo della lista realizza un elenco dei file contenuti e più richiesti da altri nodi: quelli al di sotto di una certa soglia sono automaticamente cancellati. Un'altra particolarità è che i nodi, durante il loro lavoro di inserimento e ricerca di documenti, tendono a raggruppare (clustering) questi ultimi per similitudine di chiave, ogni nuova richiesta/inserimento di un documento fatta ad un nodo particolare, il nostro nodo scambierà con esso anche informazioni riguardanti altri documenti simili presenti. Una rete così fatta tenderà quindi ad autorganizzarsi in gruppi di nodi che condividono documenti simili. La ricerca così diventa più razionalizzata e scalabile; da sottolineare il fatto che la similitudine indica una similitudine di chiavi di documenti, che possono essere comunque dissimili per argomenti, questo per evitare il raggruppamento di materiale simile su un ristretto gruppo di nodi che renderebbe la rete vulnerabile.

Configuriamo freenet

Prima di tutto identifichiamo il tipo di uso che vogliamo farne: se intendiamo fare del nostro nodo la parte server, è consigliabile, ma non strettamente necessario, avere una connessione permanente ad internet, un indirizzo IP statico e nessun firewall o proxy fra il nodo e internet. Comunque viene data anche la possibilità di navigare in freenet con le sole funzioni client.
Scarichiamo il pacchetto. Il sito di riferimento è *http://freenet.sourceforge.net,* qui troviamo il tarball, documentazione di installazione, manuali etc. Inoltre se proprio insistete, è possibile trovare il pacchetto software con le note di installazione per Windows. L'ultima versione stabile, al momento della scrittura di questo articolo, è la 0.3.9.1. Un requisito fondamentale, dato che il software per freenet è scritto in java, è l'installazione di un java runtime environment (JRE) 1.3 o superiore. Per la mia macchina ho installato il SUN-JRE, però quasi tutte le distribuzioni linux forniscono il JRE Kaffe, una valida soluzione opensource.
Per vedere la versione della nostra virtual machine impostiamo il seguente comando:

```
java -version
```

Per prima cosa esplodiamo il pacchetto:

```
tar -xvzf freenet-0.3.9.1.tar.gz
```

questo crea nella directory corrente la sottodirectory *Freenet.* In questa directory si trova il file di configurazione del server *sample.freenetrc* che bisognerà rinominare in *.freenetrc;* esso contiene le impostazioni sia del client che del server.
Vediamone qualcuna:

```
listenPort=<portaServer>
```

Questo parametro indica il valore della porta su cui il nostro server si mette in ascolto. Esso può assumere qualsiasi valore tra *5000* e *65535.*

```
logFile=freenet.log
```

Indica il nome del file di log del server.

```
nodeAddress=<indirizzo IP del nodo>
```

Questo parametro è commentato di default.
Valorizzare questo, solo se la nostra macchina è fornita di un indirizzo IP statico, allo scopo di farlo conoscere agli altri nodi per agevolarli nella ricerca.

```
nodeFile=nodes.config
```

Questo file contiene gli indirizzi e le porte di altri nodi presenti nella rete. In seguito si vedrà come reperire un gruppo di questi nodi.

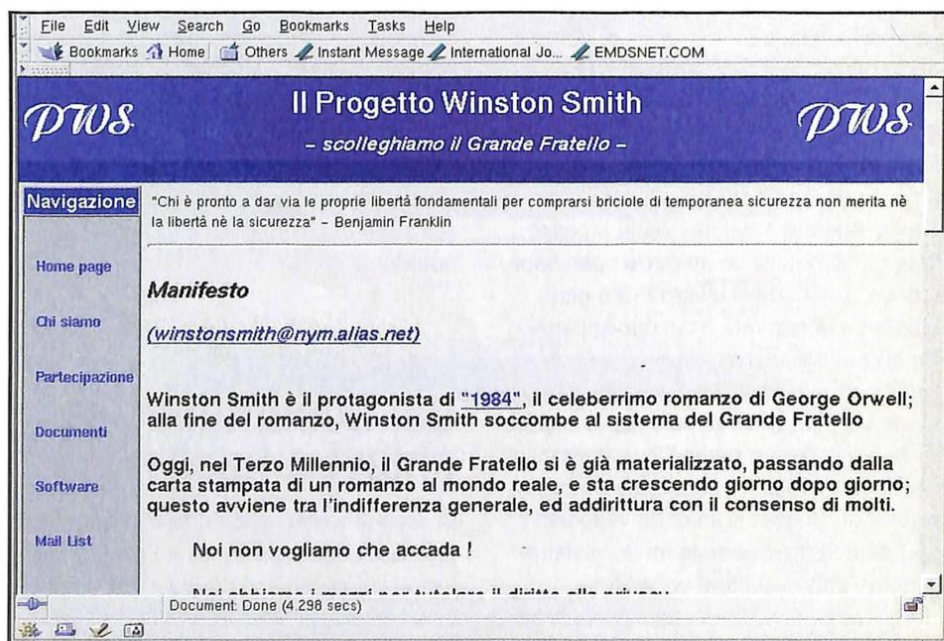



Figura 10
Il progetto Winston Smith - manifesto delle libertà digitali.

```
transient=yes
```

Questo parametro è valorizzato a *yes* (valore di default), solo per quei nodi che non funzionano da server in quanto non in connessione permanente e con indirizzo IP dinamico. Questo file può anche essere generato dinamicamente eseguendo il seguente comando:

```
java -cp freenet.jar
Freenet.scripts.Setup .freenetrc
```

A questo punto si crea il file *.proxyrc* con il seguente contenuto:

```
[FProxy]
serverAddress=127.0.0.1:<portaServer>
```

A questo punto il nostro nodo è pronto per l'avvio: lanciamo l'eseguibile

```
./freenet_server & nohup
```

in questo modo il nostro server è avviato in background (opzione *&*) e con il comando *nohup* si mantiene il processo attivo anche dopo la chiusura della shell. Un altro modo per lanciare il server è:

```
java -cp freenet.jar freenet.jar
Freenet.node.Node & nohup
```

oppure per il JRE Kaffe:

```
java -jar freenet.jar Freenet.
                   node.Node & nohup
```

Si può fermare il processo con il seguente comando:

```
killall java
```

Navigazione

Per avviare il client è sufficiente lanciare il browser preferito su questo URL: *http:// 127.0.0.1:8081* oppure *http:// localhost:8081*. Però prima di avviare la navigazione in ricerca dei documenti è opportuno andare nel sito web: *http://www.octayne.com/inform.php*. Questa pagina restituisce una lista di nodi attivi in quel momento. Inoltre il vostro indirizzo IP viene aggiunto automaticamente alla lista, quindi bisogna ricordarsi di rimuoversi dalla lista prima della fine della sessione di lavoro lanciando l'URL: *http://www.octayne.com/inform.php?remove=1*. Per aumentare la vostra possibilità di ricerca copiate, gli indirizzi di questa lista nel file *nodes.config* presente nella vostra directory di freenet; attenzione però: questo file non deve contenere linee vuote ed altre informazioni, ad eccezione degli indirizzi dei server così ottenuti, come nell'esempio:

```
<inizio del file>tcp/
                130.233.89.94:11415
(...) altri indirizzi server
tcp/80.71.237.77:19114<fine del file>
```

Ecco invece alcuni esempi errati:

```
<inizio del file>
tcp/130.233.89.94:11415
(...) altri indirizzi server
tcp/80.71.237.77:19114<fine del file>
```

oppure:

```
<inizio del file>tcp
               /130.233.89.94:11415
(...) altri indirizzi server
tcp/80.71.237.77:19114
<fine del file>
```

Osserviamo la Figura 1: questa mostra la schermata del browser sull'indirizzo *http:// localhost:8081* oppure *http://127.0.0.1: 8081* del vostro gateway. Per testare il corretto funzionamento del vostro nodo potete inerire la chiave di prova *'test-key'*, così come indicato in figura: se tutto funziona correttamente potete salvare il file test-key dal seguente contenuto:

```
'Congratulations! You've fetched a
Freenet key'
```

Ricerca delle chiavi

Occorre sottolineare che per la rete freenet non esiste (ancora) un'indicizzazione degli indirizzi simile a quella dei motori di ricerca web, in quanto la necessità principale è quella di garantire l'anonimato della fonte. Un accentramento di informazioni che rendano possibile la reperibilità dei documenti comprometterebbe questo importante requisito. Esistono comunque alcuni siti dove sono pubblicati elenchi di chiavi di documenti che possono essere immessi direttamente nel nostro gateway. Uno dei più noti è lo *Steve's Index Key* all'URL: *http://thalassocracy.org/keyindex*. Si inserisce l'argomento di ricerca nell'apposita maschera.
Ricordiamo che la ricerca viene fatta sul contenuto della chiave freenet e non sul contenuto del documento, come avviene per il web; una volta trovata la chiave si inserisce nel nostro gateway e... buona navigazione!

Gateway nel web

Freenet è, come abbiamo detto, una rete molto giovane e ancora in via di evoluzione. La ricerca dei nodi e dei documenti non è facile e può essere facile scoraggiarsi. Per facilitare la ricerca è possibile trovare dei gateway nel web. Ce ne sono due italiani: *https://freenet.autistici.org:4433/* *https://freenet.firenze.linux.it:4433/*. Il gruppo di freenet di Firenze ha anche una mailing list, a cui collabora Marco Calamari, i cui consigli sono stati molto utili per la stesura di questo articolo e a cui vanno i miei ringraziamenti. Chiudiamo con un documento interessante in materia di privacy. Ecco la sua chiave:

```
MSK@SSK@4YqXGejNtlzwoCXo23fCYe
VH~lwQAgE/20011118000000-pws//
```

Conclusioni

Freenet è uno strumento molto efficace per la ricerca di documentazione che garantisce da una parte l'autenticità del documento ricercato, dall'altra la privacy dell'autore del documento. Sicuramente per la maggior parte dei navigatori su web, non c'è, o almeno non c'è al momento, la rapidità e la comodità nel reperire informazioni. Questo è dovuto oltretutto al fatto che freenet è una rete giovanissima, infatti ha appena un anno vita, mentre l'esplosione del web si può datare a circa dieci anni fa, un divario che in informatica è tantissimo, ma che altrettanto rapidamente può essere colmato.

Marco Giangrasso

Anno nuovo... desktop tutto nuovo ...(forse)

Il pentolone delle novità per i due desktop environment che si dividono il panorama dei client Linux ribolle vistosamente. KDE e Gnome sono due tra i progetti più vivi e vitali dell'Open Source. Cosa ci riserva il prossimo imminente futuro?

A dire la verità ci eravamo appena abituati ad alcune delle novità più vistose delle ultime recenti versioni dei desktop environment che ormai sono diventati sinonimo dei client Linux: KDE e Gnome. Il KDE, dalla versione 2, era migliorato dal punto di vista grafico perdendo quell'aspetto ospedaliero che lo rendeva smunto ed emaciato come un noto sistema operativo in via di estinzione. Era diventato più veloce e finalmente evitava di fare a cazzotti con i font di sistema. L'interazione era così diventata più divertente e fluida e la risposta più immediata. Gnome dalla versione 1.4 di contro, che è sempre stato carino, divertente, variopinto e un po' (anche troppo) estroso, aveva iniziato addirittura a dare l'impressione di voler funzionare in tempi ragionevoli, lasciando persino un po' di memoria RAM a disposizione delle applicazioni utente, mentre in precedenza era necessario per l'utente fare la scelta fondamentale se usare qualche applicazione o semplicemente guardare il proprio desktop -sebbene molto bello. Ambedue però erano ben al di qua dalla linea ideale che separa la perfezione dalla mediocrità così i due team di sviluppo, smesse le chiacchiere e le fiammate post-rilascio, hanno ben pensato di tirarsi su le maniche e ricominciare a pensare alla successiva versione.

Fare un Desktop nel 2002

Il panorama in cui i gruppi di lavoro avrebbero dovuto trovarsi a lavorare erano però ben diversi da quello precedente.
Tutte le belle speranze e la baldanzosa prosopopea di Miguel de Icaza (boss dello sviluppo Gnome) e della sua Ximian si sono dissolte con le fluttuazioni di borsa e nella raccolta di fondi ben al di sotto delle aspettative e sufficienti al più a pagarsi un mediocre stipendio. Sull'altro versante, le prospettive più casarecce e pragmatiche del team KDE, spinte dal buon modello di sviluppo commerciale e nel contempo libero che la Trolltech ha saputo crearsi, ha dato al variegato e ampio mondo dei collaboratori volontari KDE la possibilità di concentrarsi più sul software in sé che sugli accessori secondari come cercarsi uno stipendio e una prospettiva di vita.

Figura 1

Gnome desktop.

SUN mangia tutto?

L'ingresso in campo della Sun Microsystem e del suo non sempre leale management, che irrompendo sulla scena dell'Open Source in una sorta di do-ut-des, ha realizzato di fatto uno scambio tra il codice di StarOffice, posto nel Free Software ma di cui comunque Sun rimane saldamente principale contributore e beneficiario, e l'ingresso in una posizione di assoluto rilievo nel gruppo di controllo dell'evoluzione di Gnome. Accertatosi della realizzazione di questo disegno Scotty McNealy ha iniziato ad investire pesantemente su Gnome come desktop environment principale del suo Unix proprietario Solaris. Il nostro ingenuo lettore potrebbe pensare che Sun forte delle proprie centinaia di sviluppatori dipendenti a programmare per colmare le evidenti carenze del software. No di certo. Ma realizzando estesi (e costosi, così dicono) test di usabilità dell'ambiente per orientarne decisamente lo sviluppo.
L'azienda californiana, a costi ragionevolmente limitati per un vendor commerciale, si trova nel punto focale di tre aree

Figura 2

Gnome desktop.

particolarmente delicate. Nel campo dei kernel Unix ha a disposizione un non disprezzabile Solaris disponibile praticamente gratis sulle proprie piattaforme RISC e su i386 pronto a fare il grande balzo nella competizione con Linux. Nel campo dei desktop environment può contare su una posizione di sicuro rilievo nello sviluppo di Gnome, e quindi di una fenomenale integrazione con il proprio Solaris. Nel campo delle applicazioni da ufficio dispone dell'ambiente StarOffice, anche come ponte verso i servizi e le applicazioni internet. Una posizione particolarmente imbarazzante per Miguel de Icaza e soci che devono oggi trovarsi a giustificare questa relazione perversa proprio quando la nascita del proprio progetto era dovuta ad una orgogliosa rivendicazione di libertà contro le librerie Trolltech, alla base di KDE, che allora non erano state rilasciate nel free-sofware. Non è strano quindi che i contorni dello sviluppo prossimo venturo di Gnome siano così nebulosi ed incerti.

La pubblicazione di una tabella di marcia che prevede un (improbabile?) rilascio finale per il 15 Febbraio sembra solo una contromossa alla rigida tabella di marcia che il team KDE si è imposto che prevede l'uscita della propria versione 3.0 solo 10 giorni dopo.

Figura 3

Un'altro screen shot del KDE 2.0.

Come ti rilascio il pupo

Non è solo la modalità di pubblicazione del calendario, ma anche la diversità dell'approccio al rilascio del software cambiano radicalmente tra i due progetti. Il KDE adotta un approccio "planning freeze" congelando la pianificazione di nuove aggiunte alla base del codice in fase di progettazione, riservandosi di arrivare in fondo alla data di rilascio con la concreta implementazione.

Si sanno quindi tutte le novità del desktop addirittura nella fase iniziale dello sviluppo. Potrebbe mancare qualcosa rispetto a quanto dichiarato, ma il team ha dimostrato finora di aver saputo mantenere sostanzialmente gli impegni. Il difetto essenziale di questo schema è che le funzionalità più ostiche da implementare, o quelle secondarie, saranno anche quelle più giovani e meno stabili della base di codice.

Nel caso di Gnome, invece, l'approccio di rilascio e un più orientativo "feature freeze". In questo caso la Beta 1 sarà la versione su cui saranno disponibili una gran quantità di nuove caratteristiche, non precedentemente pianificate che saranno via via scremate portando a termine solo quelle che... si riusciranno a terminare (il discorso di per sé non è chiaro e lascia adito a molte distorsioni). Perchè con un modello di sviluppo tale si raggiunga una certa stabilità non è certo strano dover aspettare un po', in compenso lascia molta più libertà ai programmatori di integrare nuove caratteristiche, sebbene finisca per lasciare, il più delle volte, nelle versioni Beta residui di funzionalità, semmai molto interessanti, non adeguatamente razionalizzate e sviluppate che semplicemente svaniscono nelle versioni candidate al rilascio definitivo (nella migliore delle ipotesi) o vi rimangano con un livello qualitativo decisamente carente.

KDE 3.0

Quanto alle nuove caratteristiche dei due nuovi ambienti è quindi KDE che guida l'innovazione, almeno quella pianificata. Innanzitutto KDE 3.0 sarà allineato alle librerie QT nella loro più nuova versione 3.0, già rilasciata nei tempi previsti.

Questo solo passaggio aprirà, ad un prezzo veramente ridotto per gli sviluppatori KDE, un completo nuovo insieme di interessanti possibilità. Un esempio rilevante è il

supporto nativo Unicode che permette una più efficiente ed estrema nazionalizzazione delle applicazioni, anche a livello di set di caratteri. Ci sarà inoltre il supporto per l'Alpha Blending, ovvero la gestione degli effetti di semitrasparenza. Dal punto di vista delle applicazioni KDE offrirà l'ambiente di sviluppo KDevelop 3.0 (simile ai linguaggi Visual di Microsoft o al Delphi/Kylix di Borland) che permetterà la realizzazione di applicazioni integrando anche il QT Designer per le interfacce grafiche basate sulle librerie QT e la gestione dell'accesso ai database SQL, diventando di fatto un vero e proprio sistema RAD (Rapid Application Development).
L'integrazione delle applicazioni si farà sentire anche nel client di posta elettronica *KMail* che assorbirà l'attuale *KNode* per la gestione dei newsgroup. Si promette inoltre un miglioramento generale di *Konqueror*, il browser HTTP già oggi considerato uno dei migliori prodotti disponibili per Linux, e infine l'integrazione di una piattaforma di Edutainment per lo sviluppo di applicazioni a metà tra l'educativo e il videogioco.
Infine, di interesse limitato per gli utenti, ma non per gli sviluppatori, KDE sarà finalmente programmabile non solo in C++ ma anche in Java, C e Objetive C.

Gnome 2.0

E Gnome? La versione dell'anteprima delle librerie di sviluppo rilasciata alla fine del 2001 ha lasciato molto spazio all'immaginazione (diciamo così).
Infatti non si capiva effettivamente su cosa gli sviluppatori della nuova versione sarebbero stati chiamati a cimentarsi.
La logica (un po' bacata) era che ci sarebbe stato tutto quello che c'era in KDE 3.0 e molto di più. Cosa? Di sicuro l'Unicode, l'alpha blending e l'anti-aliasing dei font (tutto già di fatto implementato dalla Trolltech alla data del rilascio della sola anteprima Gnome).
E poi? Dal punto di vista delle applicazioni ci sarà di sicuro l'integrazione del meraviglioso *Evolution*, il client di posta elettronica e gestione di informazioni personali che aspira ad essere il contraltare di Outlook. E poi ci sarà sicuramente una completa riscrittura dello *Gnome Control Center*, praticamente crollato sotto le critiche e le proteste degli utenti vista l'assoluta confusione in cui regnava tanto da renderlo più spesso dannoso che inutile. Non si sa se sarà sostituito da una versione estesa degli Ximian Setup Tools o continuerà ad avere l'attuale look-and-feel assorbendo invece le funzionalità di quest'ultimo programma.

Figura 4

KDE desktop.

Conclusioni

Per l'utente finale queste nuove versioni dei desktop non saranno un particolare salto di qualità. Sono però versioni decisamente importanti per gli sviluppatori. KDE dovrà dimostrare di aver stabilizzato la propria indipendenza come movimento di sviluppo e provare sul campo un modello di sviluppo che stante la sua giovinezza non si può certo considerare infallibile.
Ancora più grande è la sfida per Gnome, che sarà chiamato innanzitutto a non distruggere quel poco di stabilità in esecuzione che si era faticosamente conquistato con la 1.4, dovrà fare i conti con la presenza ingombrante di SUN, da un lato, e dai mal di pancia degli sviluppatori più "politicamente" orientati dall'altro. Dovrà inoltre dare una concretezza a molte delle idee che finora sono rimaste solo negli accorati appelli elettronici, ma raramente si sono tramutate in codice funzionante.
Agli user l'ardua sentenza...

Come destreggiarsi tra log e file di configurazione

Neofita assoluto? Niente paura. In questi articoli c'è tutto quello che serve per iniziare: come utilizzare, cambiare, comprendere il comportamento e personalizzare l'interfaccia a linea di comando!

Come in tutti gli articoli di questa serie, puoi provare gli esempi che daremo direttamente in una console di testo (prova a digitare *<Ctrl>-<Alt>-<F1>*, poi digita il tuo *nome utente* e la *password*) oppure da una finestra di terminale, se preferisci lavorare in modalità grafica. Nei sistemi unixoidi, a differenza che in Windows, l'utilizzo dei file di testo è estremamente ampio. Molti dettagli del comportamento del sistema sono specificati in file di testo; nella directory */etc*, ad esempio, dove è concentrata la maggioranza dei file di configurazione (e dove lo standard per il filesystem Linux raccomanda che siano installati tutti i file di questo tipo), sono presenti esclusivamente file di testo. Altri file di configurazione sono in genere presenti nelle directory di configurazione degli utenti, quelle che iniziano con il punto (e sono quindi normalmente nascoste all'output di *ls(1)*, vedi articolo nello scorso numero) nella home directory. Altri file di testo sono quelli di log, registrazioni di eventi da parte di programmi non interattivi lanciati all'avvio del sistema (i cosiddetti daemon), ma anche da parte del kernel stesso. In Linux questi file sono collocati nella directory */var/log*. I nomi e i contenuti dei diversi file dipendono dalla configurazione del sistema, ovvero innanzitutto dalla tua distribuzione: molto comuni sono i file message e debug, che concentrano buona parte dei messaggi secondo due distinti livelli di severità. Una categoria peculiare dei file di testo è formata da quelli presenti nella directory */proc*. I file e le directory ivi contenuti non sono effettivamente presenti su disco, ma sono creati al volo dal kernel quando un utente o un programma tentano di leggerli o di scriverci. Così, i file sembrano avere dimensione zero, ma ciononostante essi possono essere letti e, in qualche caso, anche scritti:

```
$ls -l /proc
dr-xr-xr-x    3 root     root
                         0 Dec  8 10:54 1
...
-r--r--r--    1 root     root           0
                    Dec  8 10:54 cmdline
-r--r--r--    1 root     root           0
                    Dec  8 10:54 cpuinfo
-r--r--r--    1 root     root           0
                    Dec  8 10:54 devices
...
$ less /proc/cpuinfo
processor       : 0
vendor_id       : AuthenticAMD
cpu family      : 6
...
cpu MHz         : 800.062
...
```

In pratica, la directory */proc* consente di accedere ai dati in possesso del kernel. Se non ti sembra strano che siano file di testo, ripensaci: i dati all'interno del kernel sono in formato binario, per cui il kernel oltre a ripescare i dati deve anche formattarli come un file di testo. Questa operazione è per sua natura inefficiente, tanto che alcuni sviluppatori del kernel hanno proposto di avere file binari in */proc* (proposta che non ha mai trovato d'accordo Linus, e che quindi fortunatamente non è mai andata avanti).

Ulteriori esempi di file di testo sono le caselle di posta (in */var/spool/mail*), i sorgenti dei programmi in qualunque linguaggio(ad esempio quelli del kernel, che trovi in */usr/src/linux* o in */usr/src/kernel**, se li hai installati), gli script per la shell o per interpreti come perl e python, alcuni formati per la creazione di documenti con formato (come html, xml, TeX, rtf), i classici formati di stampa postscript, e altro ancora.

Esempi di file non di testo sono i programmi, le librerie, i file compressi, e alcuni formati di file di dati.

```
Manual Page
Options | Sections        The current manual page is: ascii.
ASCII(7)              Linux Programmer's Manual              ASCII(7)

NAME
       ascii - the ASCII character set encoded in octal, decimal,
       and hexadecimal

DESCRIPTION
       ASCII is the American Standard Code for Information Inter-
       change.  It is a 7-bit code. Many 8-bit codes (such as ISO
       8859-1, the Linux default character set) contain ASCII  as
       their  lower half.  The international counterpart of ASCII
       is known as ISO 646.

       The following table contains the 128 ASCII characters.

       C program '\X' escapes are noted.

       Oct   Dec   Hex   Char           Oct   Dec   Hex   Char
       ------------------------------------------------------------
       000   0     00    NUL '\0'       100   64    40    @
       001   1     01    SOH            101   65    41    A
       002   2     02    STX            102   66    42    B
       003   3     03    ETX            103   67    43    C
       004   4     04    EOT            104   68    44    D
       005   5     05    ENQ            105   69    45    E
       006   6     06    ACK            106   70    46    F
       007   7     07    BEL '\a'       107   71    47    G
       010   8     08    BS  '\b'       110   72    48    H
       011   9     09    HT  '\t'       111   73    49    I
       012   10    0A    LF  '\n'       112   74    4A    J
       013   11    0B    VT  '\v'       113   75    4B    K
       014   12    0C    FF  '\f'       114   76    4C    L
       015   13    0D    CR  '\r'       115   77    4D    M
       016   14    0E    SO             116   78    4E    N
       017   15    0F    SI             117   79    4F    O
       020   16    10    DLE            120   80    50    P
       021   17    11    DC1            121   81    51    Q
       022   18    12    DC2            122   82    52    R
       023   19    13    DC3            123   83    53    S
```

Figura 1

Le pagine di manuale ascii(7), iso-8859-1(7), ecc., contengono le tabelle dei codici dei caratteri nelle diverse codifiche.

Il contenuto del file

Ma come si fa a capire se un file è un file di testo o è in formato binario? Come abbiamo mostrato nello scorso numero, è sufficiente utilizzare il comando *file(1):*

```
$ file /etc/fstab

fstab: ASCII text
```

Notare che in alcuni casi vengono dati ulteriori dettagli, ma resta sempre presente la parola chiave *text* che permette di capire che si tratta di un file di testo:

```
$ file /etc/init.d/rc
/etc/init.d/rc: Bourne shell script
                              text executable
```

Se si vuole visualizzare un file, comunque, non è necessario stabilire prima che si tratta di un file di testo, e quindi passare alla visualizzazione: ci si può anche limitare a passarli a uno dei classici visualizzatori, *more(1)* e *less(1)*. Se si esegue, ad esempio, il comando *more /bin/ls*, il programma uscirà con un messaggio di errore; invece *less*

Ma cos'è, poi, un testo?

Ricordiamo dall'articolo dello scorso numero che in Linux esistono sette tipi di file: *file regolari, link, directory, dispositivi a blocchi, dispositivi a carattere, socket* e *pipe* con *nome.* I file regolari sono quelli che più comunemente ci si aspetta di sentire chiamare *"file"*: una sequenza di byte memorizzati su un dispositivo fisico. Un file di testo è un file regolare che contiene esclusivamente caratteri stampabili. Chiaramente, questo concetto dipende dalla codifica utilizzata; se utilizziamo una codifica estesa, appartenente alla famiglia *iso-8859*, sono esclusi solo i caratteri con valore decimale compreso tra zero e 31 e quello con valore 127, cioè gli storici caratteri di controllo ASCII. In codifica ASCII pura, invece, vanno esclusi anche tutti i caratteri con valore compreso tra 128 e 255, cioè in cui il bit più significativo abbia valore 1. Ovviamente, con questa definizione, un file di testo ASCII è anche un file di testo iso-8859. Il comando *file(1)*, che riporta sempre la classificazione più restrittiva, indica anche il tipo di codifica, distinguendo i file di testo ASCII rispetto a quelli iso-8859. I comandi discussi nell'articolo funzionano con qualunque file regolare, ma sono concepiti per l'elaborazione dei file di testo. Con l'eccezione dei file di visualizzazione, non creano problemi se utilizzati su file binari.

Abbecedario di risoluzione dei problemi

Perché i visualizzatori di file sono così restii a visualizzare un file binario? Sia gli emulatori di terminale che si utilizzano in modalità grafica, sia le console virtuali che si utilizzano in modalità testuale, interpretano alcuni caratteri binari che vengono loro inviati per la stampa come comandi di configurazione. L'effetto di un comando apparentemente ingenuo come *less /bin/ls,* ad esempio, può andare dalla riconfigurazione della tastiera, che può alterare il funzionamento di tasti come quello di cancellazione e quelli di direzione, alla riconfigurazione del set di caratteri, che rende illegibile il testo contenuto sul terminale.
Emulatori di terminale più moderni come *Eterm, gnome-terminal* e *konsole* sono meno soggetti a questo problema, ma non del tutto immuni. In ogni caso, se mai doveste trovarvi "incastrati" in un terminale malconfigurato, eseguite le seguenti operazioni:

- Premete un paio di volte il tasto *invio.*
- Se non vi è stato alcun effetto, una sequenza di caratteri ha bloccato il terminale. Premete *<Ctrl>-Q,* quindi di nuovo *Invio.*
- Se di nuovo non accade nulla, utilizzate il tasto denominato *<Bloc Scorr>* o *<Scroll Lock>.* Il terminale ricomincerà a rispondere ai caratteri di *Invio.*
- Digitate il comando reset, seguito da *Invio.* Il terminale tornerà normale.

Queste operazioni possono essere eseguite anche "alla cieca", qualora il terminale avesse un set di caratteri sbagliato.
Se, nonostante questi tentativi, il terminale dovesse continuare a non rispondere, potete provare la sequenza *<Ctrl>-<Alt>-<F1>* e simili per spostarvi su un'altra console virtuale, *<Ctrl>-<Alt>-<Backspace>* per terminare la modalità grafica, oppure *<Ctrl>-<Alt>-<Canc>* per effettuare un reboot. In otto anni di utilizzo di sistemi Gnu/Linux, mi è successo una sola volta di trovare un sistema di cui non riuscivo a riguadagnare il controllo con questi trucchi (stavo tentando di forzare l'installazione di X11 con una scheda grafica non supportata).

avviserà che si sta tentando di accedere a un file binario, e chiederà se si vuole proseguire. Di questi due visualizzatori abbiamo già parlato nel n.11; qui ci limitiamo a raccomandare il secondo, ricordando che i tasti di navigazione funzionano come ci si aspetta, e che la guida on line è disponibile tramite il tasto *'h'.*
Il comando *cat(1)* (abbreviazione di concatenare) non è specifico per i file di testo ma viene più spesso utilizzato con questi. Nella forma di base, cat nomefile, non fa altro che stampare il contenuto di un file sullo schermo. La forma *cat file1 file2 file3 ...,* come ci si può attendere, concatena i file e li stampa. La forma *cat file1 file2 ... > ofile,* quindi, crea un file che è la concatenazione dei precedenti. È anche possibile prendere l'input dallo schermo: se si digita *cat > nuovofile,* ad esempio, verrà visualizzata una riga vuota in cui è possibile iniziare a scrivere il contenuto del nuovo file (che viene chiuso premendo *Invio* e quindi *<Ctrl>-D*). Questo trucco è particolarmente utile per creare un file contenente parte dell'output di un comando, presente su un'altra finestra di terminale o in un'altra console e catturato con il mouse. Naturalmente, si possono utilizzare tutti i trucchi standard della redirezione, ad esempio il piping verso un altro comando (vedi n.12) oppure il doppio segno di maggiore per appendere su un altro file: *cat file1 file2 >> vecchiofile* aggiungerà *file1* e *file2* a *vecchiofile, cat >> vecchiofile* permetterà di appendere manualmente del testo a un file preesistente, e così via.
In molti casi si è interessati solo alla parte finale di un file, ad esempio per vedere le ultime righe di un log; in questo caso, si può utilizzare il comando *tail(1).* Una classica applicazione è il comando *tail /var/log/messages* che permette di visualizzare gli ultimi messaggi di sistema. Di default vengono mostrate dieci righe; se si vuole un numero diverso di righe si può usare il parametro *-n,* ad esempio *tail -20 /var/log/messages.* Se invece si vuole "tenere d'occhio" un file mentre viene modificato, si può utilizzare il parametro *-f;* così, il comando *tail -f /var/log/messages* mostrerà la parte finale del file, aggiornandola man mano che vengono aggiunte righe. Occasionalmente utile è il comando *head(1),* che, come ci si può attendere, mostra la parte iniziale di un file. Anche questo comando mostra di default dieci linee, numero che può essere modificato con il parametro *-n;* ma non esiste l'equivalente del parametro *-f,* dal momento che generalmente i file non vengono sovrascritti dall'inizio.
È interessante notare che tail e head possono essere utilizzati insieme per selezionare una specifica linea di un file; il comando:

```
head -100 /var/log/messages | tail -1
```

stamperà la centesima riga del solito file, mentre il comando:

```
tail -2 /var/log/messages | head -1
```

mostrerà la penultima. Prova a fare questo, in Windows! Ancora più esotico è il comando *tee(1),* che stampa contemporaneamente in più posti. Un classico esempio di utilizzo è il salvataggio su file di comandi che vengono mostrati sullo schermo; i messaggi risultanti dalla compilazione del kernel, effettuata con il comando *make bzImage,* possono essere salvati su file e contemporaneamente visualizzati su schermo con il comando:

```
make bzImage | tee
            make_messages 2>&1.
```

Notare la sintassi *"2>&1"* per la redire-

zione degli errori oltre che dell'output regolare.

Ricerche ed elaborazioni

Stampare la centesima o la penultima linea può essere divertente, e in rari casi fa risparmiare tempo, ma è certamente meno utile che trovare la linea o le linee contenenti una certa parola, o una certa stringa. L'utilizzo del comando *grep(1)* per effettuare questo tipo di operazione è così comune nei sistemi unixoidi che il formato di certi file o di certi messaggi viene studiato di proposito; è il caso dei messaggi di sistema. Così, il comando:

```
$ grep pppd /var/log/messages | less
```

mostrerà tutti i messaggi provenienti dal daemon di connessione remota.Le caratteristiche di *grep(1)* sono sufficientemente estese da meritare un articolo a sé, cosa che abbiamo fatto un po' di tempo fa, nel n.8; qui ricordiamo rapidamente le più importanti, lasciando le rimanenti alle pagine di manuale. Il parametro *-v* permette di invertire la logica, stampando solo le linee che non contengono la stringa specificata; *grep -v kernel /var/log/messages* stamperà i messaggi non provenienti dal kernel. Si può utilizzare grep su file multipli, per individuare quali contengano la stringa. In luogo delle stringhe, si possono utilizzare espressioni regolari, nelle quali alcuni caratteri vengono interpretati in maniera speciale (*'.'* per "qualsiasi carattere", *'*'* per "zero o più occorrenze", *'^'* per "inizio linea", e così via). Ancora più sofisticato è il comando *awk(1)*, nelle sue diverse implementazioni. Oltre a selezionare le linee in base a criteri definibili a piacimento, *awk* le suddivide in campi e permette di effettuare elaborazioni. L'output dei comandi fornisce gli esempi migliori di input per *awk*; il seguente comando, ad esempio, somma le dimensioni dei file contenuti in una directory, escludendo le sottodirectory:

```
$ ls -l | awk 'BEGIN {tot=0} !/^d/
                {tot+=$5} END {print tot}'
```

I blocchi marcati con *BEGIN* e *END* vengono eseguiti prima e dopo la lettura del file, e nel nostro esempio inizializzano la variabile *tot* a *0* e la stampano dopo l'elaborazione, rispettivamente, il blocco centrale viene eseguito solo per le linee che non *('!')* iniziano *('^')* con la lettera *'d'*, cioè le directory nell'output di *ls*. L'operatore *"+="* indica, come in C, di aggiungere alla variabile *tot* il valore della variabile *$5*, cioè il quinto campo in cui è stata divisa la riga: nell'output di *ls*, la dimensione del file. Notare che il programma *awk* è contenuto tra virgolette semplici, e il pattern di selezione tra barre semplici. Evidentemente *awk* è più complesso di *grep*, dal momento che può effettuare, con una sintassi più complessa, esattamente le stesse operazioni; i seguenti comandi sono equivalenti agli esempi fatti per *grep:*

```
$ awk '/pppd/' /var/log/messages
                                 | less
$ awk '!/kernel/' /var/log/messages
                                 | less
```

Awk ha un insieme completo di operatori, consente di creare array di variabili, contiene varie funzioni e variabili predefinite, ha ricche funzionalità di input e output, e consente all'utente di definire le proprie funzioni. In breve, è un vero e proprio linguaggio di programmazione, che richiede un certo studio per essere padroneggiato, ma che consente anche utilizzi semplici e immediati come quello mostrato.

Il problema delle dimensioni

I file di testo si pongono, in un certo qual modo, come linguaggio di base per l'interfaccia tra essere umano e calcolatore; da qui deriva l'onnipresenza di questi file nei sistemi unixoidi. Questi file hanno però un difetto estremamente grave: sprecano un sacco di spazio! Fattori tipici di compressione raggiungibili per un file di testo sono dell'ordine di dieci; pertanto, spesso i file di testo vengono immagazzinati su disco in forma compressa, e molte delle applicazioni di nuova generazione in ambiente GNU/Linux utilizzano per i propri file un formato XML compresso. Questo pone il problema di come applicare i comandi discussi, e i numerosi altri disponibili, a file di testo compressi. La risposta sta nell'utilizzo del piping (n.12), che permette di applicare i comandi a file decompressi "al volo". Per visualizzare un "howto" nel mio sistema Debian, ad esempio, eseguo il comando seguente:

```
$ cd /usr/doc/HOWTO/en-txt
$ gzip -cd Ethernet-HOWTO.gz | less
```

Notare il parametro *'d'* che indica a *gzip(1)* di decomprimere, e il parametro *'c'* per stampare il file risultante sullo standard output anziché salvarlo (cosa che, come utente non root in una directory di sistema, non potrei comunque fare). In realtà, ho mentito: non faccio proprio così. Mi limito a dire:

```
$ gzless  Ethernet-HOWTO.gz
```

in cui *gzless* è una funzione che ho definito in */etc/zsh30/zshrc:*

```
function gzless ()
{ gzip -cd $1 | less }
```

Con bash si può utilizzare esattamente la stessa sintassi, avendo però cura di porre la definizione in */etc/bash.bashrc*.

Conclusioni

Esistono numerosi altri comandi che rientrano nella categoria della manipolazione dei testi, alcuni dei quali essenziali per compiti specifici (come ad esempio *sort*), altri facilmente emulabili tramite *awk*. I casi discussi coprono le esigenze più diffuse, mentre il solo *awk* può essere utilizzato praticamente per qualsiasi scopo.
La manipolazione dei file di testo è un'abilità praticamente essenziale per chi voglia andare al di là dell'interfaccia grafica nell'utilizzo del proprio sistema GNU/Linux.

Francesco Marchetti-Stasi

Installare la Debian

(2ª parte)

In questa seconda puntata procediamo con l'installazione della distribuzione analizzando il LILO, le fasi di reboot e l'installazione dei package

Se il sistema supporta il boot da CD basta configurarlo nel BIOS, qualora non lo sia, e inserire il primo CD della Debian, e fare un reboot. Se il computer non supporta i CD bootabili basta creare un disco di boot, con una procedura simile a quella usata in precedenza per *GNU Parted*. Trovate i file immagine in qualche sottodirectory del CD. Esercitarsi a premere *RETURN* in questi primi schermi di installazione non è un esercizio inutile. Fatelo senza dubbi. Una cosa simpatica dello script di installazione della Debian è la frequente richiesta del medium di installazione scelto. Anche quando, dopo averlo inserito un paio di centinaia di volte, pensate di essere ragionevolmente sicuri che l'abbia finalmente capito ve lo richiederà almeno un altro mezzo milione di volte. E chi crede che questo comportamento sia tremendamente noioso è in buona compagnia: certe volte fa veramente saltare i nervi. A parziale discolpa degli sviluppatori Debian va detto che questa paranoia permette setup iniziali della macchina anche in condizioni estreme di assenza di CD-Rom su reti connesse attraverso radiofari traenti della Morte Nera. Mah... andiamo avanti.

Figura 1

Quante volte vi chiederà da dove installare il sistema operativo? Tante... tante volte.

Dopo le *Release Note* sarà necessario scegliere il tipo di tastiera (Italiano *QWERTY*) e poi, la prima volta in cui bisogna porre un minimo di attenzione, rispondere alle domande riguardanti il partizionamento del disco. L'installazione continua con il programma *cfdisk* per il partizionamento. Nello spazio libero sarà necessario creare almeno due partizioni. La prima sarà una partizione Linux normale la seconda una partizione di swap. Questa partizione di swap servirà a Linux per scaricare parte dei programmi quando la memoria RAM è troppo piena. E' meglio creare prima la partizione di swap proprio dopo quella di Windows, assegnandole il *tipo 82*. Qual è la dimensione adatta? Questa è una domanda da non fare troppo in giro per evitare di essere trascinati in una delle tante guerre di religione. Logica vorrebbe di creare una partizione del doppio della memoria centrale con un limite massimo di 128 Mb. Successivamente va creata la partizione normale assegnandole il mount-point e il *tipo 83*. Altre partizioni? Forse l'unica veramente importante da creare è la */home* che vi può permettere di reinstallare eventualmente una nuova distribuzione senza perdere i vostri dati personali. Scrivere infine le partizioni e uscire da *cfdisk*.

Accettare inoltre l'inizializzazione della partizione di swap e successivamente la partizione Linux creata. Una nuova installazione non dovrà preoccuparsi della compatibilità con il *file-system 2.0*. Fare il *"bad block scan"?* Sì. Porta via un sacco di tempo e a questo punto non dovrebbe scoprire nulla di nuovo nel disco, ciò nonostante è meglio farlo subito. Montare inoltre la partizione Windows VFAT su un mount point in cui ritroveremo i file Windows ad esempio *"/windows"*. La parte noiosa dell'installazione di Debian arriva adesso: completare la configurazione di tutti i dispositivi. La cosa più semplice è trascurare tutto. Non si potrà andare in rete, non si sentirà l'audio, nè si potrà stampare. Nel caso si vogliano avere una parte di questi servizi sarà necessario importare e configurare gli appositi dispositivi.

Ad esempio per la stampante sarà necessario trovare il modulo *'lp' (Line Printer)* e avviare la configurazione che, fortunatamente, per la maggior parte possono essere auto-configurati. Scopo di tutta questa fase è ottenere un messaggio di *"installation succeeded"* su ognuno dei dispositivi che si è scelto di installare. Nel malaugurato caso in cui si ottenga invece una messaggio di *"installation failed"* sarà necessario controllare tutto ciò che riguarda l'indirizzamento di memoria o l'interrupt del dispositivo di *I/O*, informazioni che si sarebbero dovuto preparare in precedenza. Ad esempio una particolare scheda ethernet potrebbe essere descritta da una configurazione del genere: *io=0x280 irq=7*. Configurare al meglio il proprio kernel e tutti i suoi moduli è un processo abbastanza intricato che è possibile completare con il sistema in funzione. E' quindi oltre lo scopo di questo articolo. La scelta successiva dell'installazione è la semplice selezione dell'orario locale. La questione sarebbe più semplice se non ci fosse Windows che non è in grado di gestire l'orologio della macchina su GMT. Va quindi scelto di tenere nel CMOS l'ora locale (GMT-1).

```
                          cfdisk 2.10f

                     Disk Drive: /dev/hda
                    Size: 2146314240 bytes
        Heads: 128   Sectors per Track: 63   Cylinders: 519

 Name       Flags      Part Type  FS Type          [Label]      Size (MB)
-------------------------------------------------------------------------
 hda1                   Primary   Win95 FAT32                     1023.94
                        Pri/Log   Free Space                      1118.98

  [ Help  ] [  New  ] [ Print ] [  Quit  ] [ Units  ]
  [ Write ]

              Create new partition from free space_
```

Figura 2

Ecco come si dovrebbe presentare il disco dopo le condizioni ipotizzate. A questo punto potete creare le partizioni per Linux nel "Free Space".

Ma che diavolo è LILO?

Si tratta di un loader per Linux, ovvero quel piccolo programma che si occupa di accedere al disco per lanciare il sistema operativo. Anche in ambiente Windows troviamo il loader, ma è talmente ignorante che capisce solo come mandare in esecuzione Windows stesso, quindi ci toccherà sostituirlo se si vuole far partire anche Linux, a meno di non creare un dischetto di boot, nel qual caso in presenza del disco partirà Linux e senza partirà invece Windows. I loader sono piccoli programmi che risiedono in uno speciale settore del disco chiamato *MBR (Master Boot Record)*. Succo del discorso è che sarà quindi necessario chiedere al processo di installazione di inserire *LILO* nell'*MBR*.
Inoltre è bene creare un floppy di boot.

A questo punto va estratto il CD dal drive e permettere il reboot della macchina.

Reboot

La ripartenza è segnata da una ulteriore parte dello script d'installazione che permette la creazione degli utenti in cui, oltre al classico utente root va creato almeno un altro utente a basso privilegio. Il motivo principale non è tanto la sicurezza del vostro sistema in Internet, quanto per evitare i rimbrotti degli altri utenti quando vi collegherete alla posta elettronica o peggio all'IRC. Scegliete un meccanismo di protezione delle password, meglio le shadow password. E successivamente una password per root. Ricordatevela oppure scrivetela, sennò dovete rifare l'installazione. Infine c'è la possibilità di occuparsi della configurazione delle schede *PCMCIA* che vi servirà solo se state installando su un laptop attraverso un CDROM connesso ad una scheda del genere.

Selezionare ed Installare i Package

A questo punto avete messo su il sistema ma non avrete applicazioni con cui lavorare. In Debian le applicazioni sono suddivise in archivi contenenti i programmi, le librerie di supporto, i file di configurazione e la documentazione, registrati in modo compatto con informazioni sulle dipendenze da altri pacchetti. La gestione dei pacchetti è effettuata tramite il programma *dpkg* e la sua interfaccia chiamata *apt.* Attraverso *apt* e *dpkg* sarà possibile in modo abbastanza intuitivo riempire l'installazione con tutti i programmi che servono. *Apt* è così vitale che merita un lungo discorso a parte che sarà fatto nel prossimo articolo dedicato alla Debian. Scegliendo una installazione 'simple' si eviterà la precedente noiosissima fase di selezione manuale dei pacchetti tramite *dselect* e si userà piuttosto un procedimento semplificato. In questa fase più pacchetti si selezionano più saranno le singole configurazioni che si dovranno fare. Per la stragrande maggioranza dei pacchetti *apt* e *dpkg* gestiranno autonomamente la procedura di configurazione, ma alcuni sottosistemi particolarmente ostici avranno bisogno di intervento manuale. Probabilmente il più importante di questi è il sottosistema di gestione delle finestre *X-Window* e il desktop *Gnome.* Questi due sistemi permetteranno di avere a disposizione un comune desktop grafico con tutta una serie di programmi semplificati in stile Windows (e anzi molto migliori). Dopo il partizionamento del disco la corretta configurazione di *X* è il passo più complicato da compiere. Le prime configurazioni del sistema *X* sono ancora ricordate con terrore dai primi utilizzatori di Linux. Sempre con la paura che il monitor trovasse una frequenza tanto alta da farlo scoppiare, a provare e riprovare cambiando l'infinito file di configurazione facendo le prove al buio e senza un minimo di messaggi d'errore significativi. Ciò nonostante oggi tutto è estremamente semplificato dall'uso di appositi programmi di autoconfigurazione. Se si è in possesso di tutte le informazioni necessarie o di un monitor direttamente supportato si può tranquillamente limitarsi a seguire le scelte predefinite ed eventualmente fare un paio di prove per ottenere il migliore risultato possibile. Una volta acquisite tutte le informazioni inizierà la copia vera e propria dei pacchetti sul disco e i relativi processi d'installazione. A seconda della quantità scelta questo durerà da meno di dieci minuti ad oltre un ora.

Figura 3

Anche la selezione dei gruppi di package è diventata molto più semplice nell'ultima versione di Debian. Basterà selezionare i sottogruppi di cui si vuole l'installazione.

Conclusioni

Congratulazioni! Il lungo processo di installazione della Debian è adesso finito. Se avete risposto correttamente a tutte le tante domande che vi sono state poste avrete un sistema perfettamente operativo e funzionante, altrimenti alcune sue parti saranno non operative e necessiteranno di ulteriori configurazioni più raffinate. Nella prossima puntata: come migliorare la nostra configurazione, l'installazione di ulteriori pacchetti e come tenere ordine nel sistema.

XORfbServer Condivisione del desktop 0

Il protocollo RFB (Remote FrameBuffer) è quello utilizzato da VNC per la condivisione remota del desktop di una postazione di lavoro. Il nome dell'applicativo non deve ingannare circa la sua presunta complessità. Il software in questione permette, infatti, di condividere con estrema semplicità il desktop numero 0 (tanto caro agli utenti di Windows) aggiungendo così una utilissima funzionalità rispetto a quanto consentito da VNC.

Nel numero 14 della rivista abbiamo affrontato in maniera dettagliata il modo di condividere il desktop di una postazione remota utilizzando il server *VNC (Virtual Network Computing)* messo a disposizione dalla AT&T *(http://www.uk.research.att.com/vnc/)*. In quell'occasione si parlò apertamente della possibilità di condividere il desktop di computer tra loro molto distanti che potevano anche eseguire sistemi operativi diversi. Si parlò anche delle innumerevoli piattaforme supportate senza però specificare che la parte server del progetto VNC è stata

realizzata tenendo bene a mente le caratteristiche della piattaforma su cui doveva essere eseguito il server. Questo significa che se si tenta di collegarsi ad un server VNC che gira su una macchina Windows o Machintosh, ci verrà proposto quello che viene comunemente chiamato desktop numero 0 (quello principale, anche scritto come :0), mentre se si tenta di collegarsi ad una macchina Linux o Unix in generale, ci verrà proposto il primo desktop disponibile (di solito :1).
Su Linux, questo non è un problema, anzi un vantaggio; si possono infatti eseguire le applicazioni con l'opzione *-display host:n*, che consente di visualizzare l'applicazione sul desktop *n* della macchina host. Se tutto fosse ben configurato, si potrebbero eseguire le applicazioni da condividere sul desktop che viene messo a disposizione dal server VNC e realizzare quindi una postazione di lavoro remota contenente solamente quello di cui si ha veramente bisogno.
Spesso però gli utenti che utilizzano questi strumenti vengono dal mondo Windows dove l'esecuzione di una applicazione grafica è legata alla visualizzazione sul desktop del suo output grafico. Collegarsi da remoto ad una macchina Windows che sta eseguendo, che so, un programma grafico simile a xclock, e non vedere l'orologio far scorrere i secondi, può far pensare che qualcosa non funzioni correttamente. La realtà quotidiana mi fa capire che questo comportamento, del tutto normale per gli utenti Linux, non è ben accettato dagli utenti Windows.
Un certo *Jens Wagner* ha pensato a tutto questo ed ha scritto una funzionalità aggiuntiva per VNC (in realtà è un vero server VNC dato che utilizza lo stesso protocollo), che permette la condivisione del desktop numero 0, rendendo di fatto il server molto simile a quello che viene distribuito per Windows o Macintosh. Il package include diversi progetti e fornisce alcuni strumenti necessari per lavorare con il protocollo RFB. Oltre al server *x0rfbserver* di cui parleremo nel seguito di questo articolo, possiamo trovare nel package un player per stream di dati basati sul protocollo *RFB (xplayfbs)* e un visualizzatore di desktop remoti con funzioni di registrazioni macro e cattura video *(xrfbviewer)*.
L'anello di congiunzione tra *VNC* e *x0rfbserver* è proprio il protocollo utilizzato da *VNC: RFB (Remote FrameBuffer)*. Infatti, grazie al fatto che *x0rfbserver* è stato scritto utilizzando questo protocollo, possiamo utilizzare il visualizzatore di VNC per collegarci al desktop numero *0* della macchina remota. Le potenzialità dello strumento vengono ulteriormente incrementate se pensiamo al fatto che il visualizzatore di VNC non necessita di installazione ed occupa poche centinaia di KB (risiede tranquillamente su un dischetto).

Figura 1

Il sito della AT&T sul Virtual Network Computing.

Download e Installazione

Il sito ufficiale da dove poter scaricare il software è: *http://www.hexonet.de/software/rfb* dove si possono trovare i sorgenti, i binari precompilati per piattaforme i386 e le librerie necessarie per l'installazione. Purtroppo però, le istruzioni per la compilazione e l'installazione non sono più in linea quindi bisognerà arrangiarsi un po' da soli. In qualche caso su alcune piattaforme sono stati riscontrati degli errori in fase di compilazione ed è per questo che vi invito a scaricare il pacchetto, in formato rpm, direttamente dal sito *http://www.rpmfind.net* digitando nella casella di ricerca la parola *"rfb"* (ovviamente senza virgolette!). Per l'installazione sono necessarie le librerie *xclass* versione 0.5.4 o superiore (necessaria per

Figura 2

Il sito ufficiale da dove poter scaricare VNC: http://www.hexonet.de/software/rfb.

rfb) e *libxclass0* versione 0.5.4 o superiore (necessaria per *xclass*) che forniscono un ambiente grafico simile a quello utilizzato dalle applicazione Windows. Entrambe queste librerie sono scaricabili dal sito di *rpmfind* digitando come stringa di ricerca proprio il nome della libreria.

Una volta scaricati i tre pacchetti (*libxclass0, xclass* e *rfb*), per esempio nella directory */tmp*, è possibile installarli utilizzando un tool grafico per la gestione dei pacchetti come *kpackage* o *gnomerpm*. L'unica cosa da ricordare è l'ordine di installazione che deve essere necessariamente seguito per via dei controlli di dipendenza tra i vari pacchetti. Per chi, invece, non avesse a disposizione i tool di gestione dei pacchetti sopracitati è possibile aprire una finestra di terminale ed installare i pacchetti nel seguente ordine: prima *libxclass0,* poi *xclass* ed infine il *package rfb*. Diamo quindi i seguenti tre comandi:

```
# rpm –ivhlibxclass0
        <versione>.rpm

# rpm –ivh xclass
        <versione>.rpm

# rpm –ivh rfb
        <versione>.rpm
```

Configurazione e utilizzo

Il server del package rfb che permette la condivisione del desktop numero *0* si trova ora nella cartella */usr/bin* e si chiama *x0rfbserver*. Per eseguirlo basta dare il comando

 # /usr/bin/x0rfbserver

C'è poco da dire su questo utilissimo e altrettanto semplice strumento.

Dopo la sua esecuzione, si aprirà una piccola finestrella che al clic del mouse con il tasto destro aprirà un menu contestuale per la gestione del server. Sarà possibile disconnettere i vari client, disabilitare il movimento del mouse, decidere quale desktop mettere a disposizione (per default il numero 0), ma anche impostare una password che sarà richiesta al client remoto al momento della connessione. Lo strumento per connettersi al server, come detto in precedenza, può essere tranquillamente il visualizzatore di VNC. È anche possibile eseguire più istanze del server, rieseguendo il comando, per mettere a disposizione dei client desktop diversi con proprietà diverse (quello che si poteva fare anche con VNC).

Infine, una nota sul funzionamento di questa applicazione.

I test sono stati effettuati su una Linux box con installato Mandrake 8.0. ma non dovrebbero esserci problemi a trasportare tutti i concetti descritti su altre piattaforme. Inoltre, sul sito *http://www.rpmfind.net*, è possibile controllare di quali librerie necessitano i package descritti in precedenza; viene ovviamente fornita anche la possibilità di scaricare ed eventualmente installare tali librerie.

Conclusioni

A parte qualche errore verificatosi durante la compilazione dei sorgenti, la suite, una volta installata grazie agli rpm, ha mostrato un buona stabilità ed ha permesso di condividere da remoto anche il movimento del mouse (opzione ancora sperimentale) senza nessun tipo di problema.

Il protocollo su cui si basa, ereditato da VNC, permette inoltre di aggiornare lo schermo grafico del visualizzatore con estrema semplicità grazie alle pochissime richieste che devono essere effettuate al server.

Questo garantisce una maggiore velocità durante la rappresentazione di schermate grafiche anche complesse.

La speranza è che lo sviluppo della suite non sia abbandonato ma che continui a stare al passo con gli aggiornamenti costanti e regolari distribuiti da VNC.

Carlo Vasselli

Java 2 Runtime Environment, Standard Edition

"Java è il linguaggio di programmazione sviluppato da Sun Microsystems progettato per rompere i vincoli tradizionali tra l'applicazione e il sistema operativo: con Java le applicazioni possono essere eseguite su qualsiasi piattaforma hardware, così come i documenti World Wide Web possono essere visualizzati da qualsiasi browser"

Java è basato sulla potenza delle reti e sull'idea che lo stesso software può essere eseguito su parecchi differenti tipi di computer, e dispositivi di ogni tipo (il concetto di *write once, run everywhere,* ovvero scrivi una volta ed esegui ovunque). Dalla sua prima uscita del 23 maggio 1995, Java è cresciuto molto in popolarità diventando in breve tempo molto più di un linguaggio. La definizione di Java Technology, infatti, equivale oggi ad un nuovo modello per l'elaborazione dei dati, centrato sul concetto di rete e non più sul Personal Computer. Questo approccio orientato alla gestione della rete, realmente indipendente dalla piattaforma, permette alle aziende risparmi fino all'80% dei costi totali di ogni stazione di lavoro. Basti pensare allo sviluppo delle applicazioni e alla loro messa in esercizio, senza la preoccupazione derivante dai problemi di compatibilità, sempre esistiti, tra sistemi operativi e hardware disponibile. Inoltre, la possibilità di eseguire il codice senza rischi per i dispositivi di destinazione, ha contribuito a migliorare la sicurezza della rete stessa. Le principali caratteristiche del linguaggio sono le seguenti:

General purpose: può essere utilizzato per lo sviluppo di applicazioni complesse, eseguibili sia su singoli computer che in

ambiente di rete, ma anche per lo sviluppo di piccole applicazioni, dette applet, eseguibili direttamente all'interno di un browser web per aggiungere interattività alle pagine.

Orientato agli oggetti: un oggetto fa parte di una classe di oggetti da cui può ereditare il codice comune della classe.
Portabile: lo stesso software può essere eseguito su qualunque dispositivo e il linguaggio è realmente indipendente dalla piattaforma.
Multithreading: gestisce in maniera nativa l'esecuzione concorrente del codice.
Robusto: gli oggetti del linguaggio non possono contenere riferimenti (i puntatori del linguaggio C) a dati esterni a sé stessi o ad altri oggetti.
Facile da imparare: relativamente al linguaggio da cui ha preso in prestito numerose caratteristiche, il C++, è molto più didattico e facile da apprendere.

Ma come è stato possibile rendere il linguaggio indipendente dalla piattaforma? L'idea è semplice: il codice Java quando viene compilato produce una sequenza di byte che Sun Microsystems ha definito bytecode. Questa sequenza viene poi interpretata dalla *Java Virtual Machine (JVM™)*, un dispositivo di calcolo astratto capace di interpretare il bytecode. La macchina virtuale, cuore di tutto il meccanismo, è dotata di un proprio set di istruzioni macchina (dipendenti dalla piattaforma) con cui istruisce il dispositivo hardware per l'esecuzione dei comandi contenuti nel bytecode. La diffusione di massa che ha avuto il linguaggio deriva dal fatto che Netscape capì subito l'enorme potenzialità dello strumento e decise subito di introdurre nel proprio browser la JVM. Tutti gli altri principali browser presenti sul mercato si adeguarono rapidamente per permettere l'esecuzione del bytecode. Fanno eccezione la versione di Internet Explorer fornita con Windows XP (per ovvi motivi...) e i browser come Konqueror che nascono per essere strumenti multi-purpose. Per gestire queste situazioni, Sun Microsystems mette a disposizione la Java Virtual Machine come parte del Java Runtime Environment, Standard Edition (JRE), disponibile per il download all'indirizzo *http://java. sun.com/j2se/ 1.3/jre/*.

Figura 1

L'architettura della piattaforma Java 2 versione 1.3.

Download e Installazione

Dopo questa breve ma necessaria descrizione del linguaggio Java, passiamo a vedere come è possibile risolvere i problemi derivanti dalla mancanza della JVM o da una versione non troppo aggiornata.
Scarichiamo dal sito della Sun Microsystems indicato precedentemente la versione per Linux del JRE nel formato binario auto-estraibile (scegliendo l'opzione *GNUZIP Tar Shell Script*). Questo formato ci permetterà di scegliere in quale directory effettuare l'installazione. Il nome del file da scaricare sarà della forma:

jre-<numero versione>-linux-i386.bin

dove il numero di versione è, al momento della scrittura di questo articolo, *1_3_1_01* corrispondente alla piattaforma *Java 2* versione *1.3.1 (rilascio 01)*. Questa versione, così come le successive, includono anche il Java Plug-in, che descriveremo meglio in seguito. Accertiamoci di avere a disposizione i requisiti minimi e cioè: versione del *kernel 2.2.12, glibc versione 2.1.2-11* (verificabile con il comando *ls /lib/lic-**), 32 MB di RAM (raccomandati 48 MB), 16 bit di profondità di colore, KDE e KWM come windows manager, 45 MB di spazio libero sul disco di installazione. A questo punto, dopo aver verificato l'esatta dimensione del file scaricato, creiamo una directory in un punto del file system in cui abbiamo i permessi di scrittura. Se si vuole mettere a disposizione di tutti gli utenti l'ambiente runtime, possiamo tranquillamente effettuare il login come utente root e creare la cartella dentro */usr/local* con il comando:

```
# mkdir /usr/local/java
```

Posizioniamoci poi al suo interno con il comando:

```
# cd /usr/local/java
```

Ora, supponendo di aver scaricato il file nella cartella */tmp* copiamolo dentro la cartella */usr/local/java* con il comando:

```
# cp /tmp/jre-<numero
  versione>-linux-i386.bin
        /usr/local/java
```

Prima di eseguire lo script dobbiamo però renderlo eseguibile con il comando:

```
# chmod a+x jre-<numero
  versione>-linux-i386.bin
```

ed infine possiamo procedere con l'esecuzione vera e propria impartendo il comando:

```
# ./jre-<numero versione>-
          linux-i386.bin
```

Verrà visualizzata una licenza e una volta accettata verrà creata la cartella */usr/local/java/jre<versione>* nel quale si installerà il JRE.

Una volta terminata l'installazione possiamo, se vogliamo liberare un po' di spazio sul disco, eliminare il file scaricato con il comando:

```
# rm –f jre-<numero
    versione>-linux-i386.bin
```

Tutto qui. Siamo ora in grado di eseguire applicazioni Java e visualizzare applet all'interno dei browser, utilizzando la JVM */usr/local /java /jre<versione>/bin/java.*

Il Java Plug-in

Il Plug-in messo a disposizione durante l'installazione del JRE è uno strumento opzionale per i browser web che permette, però, di eseguire il codice delle applet contenute nelle pagine web utilizzando una Java Virtual Machine più aggiornata di quella installata per default al loro interno. Quindi, anche se nei browser web utilizzati risultasse già presente una JVM, la configurazione del Plug-in è da considerarsi sempre un'operazione conveniente. Con il Plug-in, infatti, sarà possibile l'esecuzione delle applet scritte con una versione del linguaggio non supportata dalla JVM contenuta all'interno dei browser e quindi si potranno sempre sfruttare tutte le nuove caratteristiche del linguaggio introdotte con le ultime versioni. Per costringere i browser web ad utilizzare il Plug-in è necessario modificare alcuni tag nel codice Html delle pagine contenenti le applet. Sun Microsystems ha pensato anche a questo. Non dobbiamo infatti preoccuparci di convertire a mano le pagine Html che devono essere visualizzate con il Plug-in. È sufficiente scaricare il convertitore dal sito Sun ed indicare quali file convertire. Dobbiamo però evitare di fare confusione con i numeri delle versioni, dato che la versione del convertitore deve essere uguale a quella del Plug-in installato. A partire dalla versione 1.3.1 del JRE (e quindi del Plug-in che, ricordiamo, è in esso contenuto), non è più possibile scaricare solamente il convertitore. Sun Microsystems ha deciso di includerlo nella suite *J2SE SDK* che mette a disposizione oltre al JRE opportuno anche il compilatore e il debugger Java. Non resta quindi che scaricarsi ed installarsi tutta la suite, procedura che affronteremo nei prossimi numeri della rivista. Se invece la versione è la *1.3, 1.3.0* oppure *1.3.0_01* possiamo trovare il convertitore sul sito *http://java.sun. com/products/plugin/1.3/converter.html* e seguire tutte le istruzioni per la sua installazione e configurazione. La documentazione messa a disposizione sul sito è comunque esauriente, quindi non lasciatevi impressionare da tutti questi numeri. Il debugging delle applet. Spesso gli errori provocati dalle applet in esecuzione non sono gestibili facilmente dato che non sempre si riesce a capire quale errore si è verificato. Con l'installazione del JRE viene però messo a disposizione anche un utile strumento di debug che permette di verificare la corretta esecuzione delle applet. Nella cartella in cui è stato installato il JRE possiamo trovare lo script eseguibile ControlPanel che, tra le tante opzioni, consente di selezionare l'opzione 'Show Java Console', molto utile per risolvere problemi apparentemente più complessi.

Configurazione di Konqueror

Konqueror non ha al suo interno una JVM per l'esecuzione delle applet Java. Bisogna quindi indicargli dove si trova quella appena installata. Nel menu *Impostazioni,* scegliamo la voce *Configura Konqueror...* Si aprirà una finestra con un menu grafico nella colonna di sinistra dove dobbiamo scegliere l'icona *Navigatore Konqueror* e poi posizionarci nella scheda *Java.* Selezioniamo tutte le opzioni e indichiamo il percorso della JVM che relativamente all'installazione appena effettuata sarà */usr/local/java/versioneJRE/bin/java.*

Configurazione di Netscape

Netscape invece è stato pensato per offrire pieno supporto alle applet Java e contiene al suo interno una JVM che potrebbe, però, essere un po' datata. Per configurare Netscape ad utilizzare il Plug-in bisogna prima eliminare le tracce di un'eventuale vecchia installazione con i comandi:

```
# rm –fr $HOME/
        .netscape/java
# rm $HOME/.netscape/
        plugins/javaplugin.so
```

poi esportare la variabile *NPX_PLUGIN_PATH* con il comando

```
# export
NPX_PLUGIN_PATH=
    <jre>/plugin/i386/ns4
```

Nel caso di Netscape 6 bisogna invece creare,al posto della variabile, un link simbolico come segue:

```
# ln -s
<jre>/plugin/i386/ns600/
        libjavaplugin_oji.so
        <netscape>/plugins
```

A questo punto, dalla finestra di terminale in cui sono state impostate le variabili d'ambiente, avviamo Netscape, chiudiamolo e riavviamolo una seconda volta (questa strana procedura si rende necessaria per via di un bug riscontrato nel browser).
Adesso, nelle preferenze di Netscape (menu *Edit/Preferences*) possiamo abilitare l'opzione relativa all'utilizzo del Java Plug-in.
Nella procedura appena descritta, prima di eseguire i comandi, bisgona ricordarsi di sostituire *<jre>* con la directory in cui è stato installato il JRE e *<netscape>* con quella in cui è installato Netscape.
Infine, per evitare di dover ogni volta reimpostare la variabile d'ambiente in una finestra di terminale, è consigliabile creare un semplice script di shell che imposta la variabile e subito dopo esegue il browser.

Conclusioni

Difficile aggiungere qualcosa per descrivere la rivoluzione voluta da Sun Microsystems. Da quando Java è stato rilasciato nel "lontano" 1995, il mondo dell'informatica è cambiato. In meglio.

Carlo Vasselli

Il "nuovo" Windows Xp vs. Linux

In questo articolo si analizzeranno le differenze fra il nuovo sistema operativo di casa Microsoft e Linux. Si analizzerà il riconoscimento dell'hardware, la semplicità d'installazione, la configurazione e l'uso nel lavoro di tutti i giorni e compatibilità con le applicazioni precedenti.

Xp è finalmente su tutti gli scaffali dei negozi d'informatica. Un confronto con Linux era d'obbligo, anche se confrontare due cose così diverse non è certo una cosa semplice e le considerazioni riportate non possono che essere il frutto di una visione certamente non approfondita e completa. Ciò nonostante, anche ad un livello di dettaglio così limitato, questo articolo può dare al lettore una utile sensazione sulle diversità di funzionamento di questi due sistemi operativi così diversi eppure così simili.

Qual è giusto per il mio computer?

Primo passo: capire se l'hardware del PC è compatibile. La prima cosa che un sistema operativo deve fare è quella di riconoscere le periferiche e i componenti e configurare di conseguenza il proprio funzionamento.
Ed ecco le prime differenze. Infatti, sebbene si dica che Linux troverebbe spesso problemi nel riconoscimento delle periferiche installate, si è riscontrato invece che la distribuzione Suse presa in considerazione abbia riconosciuto senza problemi tutte le periferiche e i componenti (scheda video, scheda audio integrata, stampante, monitor, modem esterno, mouse, CD e masterizzatore) tranne il modem interno e lo scanner. In questo caso sarà necessario affidarsi agli appositi software in grado di riconoscere il particolare scanner o modem in dotazione dell'utente. Se si eccettua questi componenti secondarie, configurabili a prezzo di un po' di sforzo e di ricerche in Internet e nella estesa

documentazione a disposizione, si può proprio dire che, purché il kernel ne preveda l'utilizzo, ogni periferica viene direttamente e facilmente autoconfigurata. Discorso a parte per Windows Xp. Di questo nuovo sistema operativo, mamma Microsoft vanta il più efficiente sistema di riconoscimento dell'hardware esistente sul mercato e pretenda che i componenti o le periferiche vengano installate automaticamente non necessitando neanche di driver specifici dei relativi produttori per funzionare al primo tentativo. Nonostante ciò non sono rari casi di strani malfunzionamenti, per esempio alcuni lettori CD rom marchiati Asus non riescono a funzionare alla velocità massima supportata. Comunque, un ottimo modo per controllare se proprio l'hardware è supportato dal nuovo sistema operativo, è quello di visitare il sito del produttore oppure di controllare con una utility integrata del CD-ROM d'installazione, se ci sono eventuali problemi di compatibilità, è nel caso, come risolverli (anche se i consigli spesso si limitano a rimandare al sito del produttore dell'hardware per ricercare driver specifici). Comunque bisogna dire, che nella maggior parte dei casi è sufficiente recuperare i driver che funzionavano sotto Windows 2000; infatti, anche se non ufficialmente certificati, questi solitamente non danno problemi.

Risorse hardware

Linux non ha tante pretese hardware. Infatti il kernel di Torvalds può essere personalizzato e ricompilato per tenere conto proprio dell'hardware presente sulla macchina su cui deve "girare". Inoltre, ha una gestione delle risorse (disco rigido e sottosistema memoria-processore) molto più efficiente dei sistemi operativi nati per l'utilizzo domestico. Con Linux non è irrealistica (sebbene decisamente limitata) una configurazione funzionante con un processore 486 e anche solo 4 Mb di memoria centrale e meno di 100 Mb di disco rigido, sebbene per un ambiente di lavoro efficace sarà necessario almeno un Pentium (uno qualsiasi anche a 75 Mhz), 16 megabyte di RAM e un disco di 300 Mb. Discorso notevolmente diverso per Windows Xp. Infatti, dire che il nuovo nato di casa Microsoft sia decisamente avido di risorse è poco. A cominciare dallo spazio richiesto sul disco rigido che supera abbondantemente il gigabyte solo per l'installazione; senza peraltro possibilità di modifiche, dato che non si possono neppure scegliere le componenti da non installare, come almeno parzialmente era in precedenza. Anche per quanto riguarda RAM e processore è necessaria una configurazione decisamente potente. Infatti, la configurazione che ci sentiamo di consigliare è una CPU a non meno di 600-800 MHz, e non meno di 128 MB di RAM per chi non faccia uso di più applicazioni contemporaneamente. Per chi invece, lavora spesso in multitasking sono vivacemente consigliati almeno 256 MByte.

Figura 1

L'interfaccia di Linux (Suse), ormai è bella come quella di Win... se non ancora più bella!

Compatibilità con Software Precedenti

Installato il S.O. e riconosciuto l'hardware, è possibile vedere come i due sistemi operativi si comportano in relazione all'utilizzo delle applicazioni utilizzate con le precedenti versioni del sistema operativo nel proprio lavoro quotidiano. Per ciò che concerne Linux, non vi sono problemi. Tutte le applicazioni delle precedenti versioni funzionano perfettamente. E le nuove eventuali versioni sono sempre compatibili all'indietro. Neppure il cambiamento di versione del kernel ha provocato sostanziali problemi alle applicazioni non ricompilate appositamente. Con Windows Xp, la Microsoft promette una più estesa compatibilità con tutte le applicazioni che prima avevano problemi con Windows 2000. Però Xp si basa sullo stesso sottosistema che precedentemente veniva messo a disposizione dei soli utenti Professional, quindi per quanti sforzi siano stati fatti per renderlo compatibile, molte applicazioni risultano tuttora non funzionanti - e presumibilmente non lo diventeranno mai più senza un pesante lavoro di migrazione del sorgente. Inoltre c'è da considerare a parte il problema dei videogiochi. Anche in questo caso, infatti, le dichiarazioni commerciali non sembrano essere basate su prove di fatto, come ciascuno può facilmente constatare. Si è detto che Windows Xp sia pienamente compatibile con tutti i giochi in commercio ma, da prove effettuate su campo, sembra che gli stessi giochi che non funzionavano sotto Windows 2000 continuano ad avere problemi sotto Xp, in compenso, però, le

Figura 2

La nuova interfaccia di Win Xp; sembra una carta di caramella, per quanto è colorata, ma può anche essere cambiata.

prestazioni ottenibili sembrano nettamente superiori.

Installazione

La procedura d'installazione di Xp è identica a quella di Windows 2000.
Dall'inizio alla fine all'utente sono mostrate solo semplici e limitate scelte, il sistema di installazione determinerà autonomamente la configurazione della macchina o al limite si bloccherà impedendo l'effettiva installazione, in asssenza delle condizioni minime di installazione.
Tale processo non concede alcun controllo su quello che verrà effettivamente installato. Che si sia utenti esperti o non si conosca nulla del computer, non si può scegliere che cosa fare o cosa installare sul proprio PC. Windows Xp è una scatola nera da prendere in blocco (o rifiutare). In un mondo sempre più orientato a lasciare agli utenti ogni singola scelta e ogni possibile variazione delle proprie configurazioni questo comportamente può risultare molto fastidioso, soprattutto se si pensa che del Gigabyte di software installato se ne userà solo una parte veramente minima, e il resto sarà non solo inutilizzato ma anche potenzialmente pericoloso perchè attaccabile da virus.

On the road... again

Mentre le interfacce grafiche Linux migliorano di giorno in giorno integrando via via strumenti di interazione utente sempre più espressivi e potenti, Windows Xp è evidentemente stato progettato sulla scia dei suoi predecessori. La filosofia che sta alla base di tutto, infatti, è la facilità di utilizzo soprattutto per gli utenti davvero le prime armi. Forse non è veramente mai stato così semplice usare un computer, ed Xp è, probabilmente, la cosa migliore che una persona che non abbia praticamente mai visto un computer possa trovarsi di fronte. Ma già per gli utenti un po' più smaliziati di Windows il comportamento di Xp è tremendamente fastidioso. E' stata scelta una nuova (e ovviamente obbligatoria) disposizione degli strumenti. Molte cose, scelte, opzioni, menù, icone, risultano leggermente modificate, e alcune procedure, ormai entrate nel bagaglio quotidiano degli utenti, devono essere riviste. Questo sacrificio sull'altare della facilità di utilizzo risulta tanto più insopportabile perchè non è modificabile in alcun modo e si ha la sensazione che Microsoft pretenda che i propri utenti migliori, invece di imparare con il tempo a fare le cose più velecomente ed efficacemente, debbano versione dopo versione essere sempre più rallentati ed impediti anche nelle cose che hanno già imparato a fare con una certa destrezza. Se è vero che una delle famose leggi di Murphy stabilisca che ogni nuova versione di qualunque software occuperà più spazio, sarà più lenta e molto meno stabile della precedente, sembra che il gigante di Microsoft voglia aggiornarla aggiungendo anche che i suoi utenti diventeranno con il tempo più idiioti.
Un comportamento che stride notevolmente con quanto accade sotto Linux e Unix in generale, dove tutte le esperienze accumulate nel tempo non risultano mai perdute tanto da permettere agevolmente ad un utente che abbia imparato ad usare Unix 30 anni fa a continuare a farlo praticamente senza dover reimparare nulla da capo, e nel contempo integrare sempre nuove e potenti modalità interattive in grado di crescere assieme alle competenze dell'utente stesso.

Stabilità senza confronto

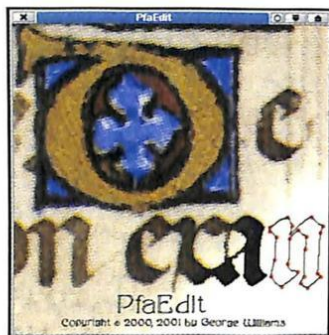
Anche dopo il notevole passo avanti in termini di stabilità di Windows 2000 (di cui Xp ha ereditato i pregi, perfezionando qualche difetto) non si può dire che Linux abbia perso lo scettro di sistema operativo più stabile. E' vero che le ormai "mitiche" schermate blu di Windows '98 sono un episodio talmente raro da non destare particolari problemi anche per un utente che richieda un'affidabilità "semi-professionale", comunque con la loro aleatorietà bisogna comunque farne i conti. Così come la gestione non senza pecche del file-system ancor oggi ben lontana dal garantire completa sicurezza ai dati utente. Il kernel Linux, invece, abitua ad una stabilità operativa senza paragoni, che gli deriva direttamente da uno UNIX adatto a compiti ben più critici di un uso domestico o semi-professionale. Meno forte, invece, è il versante delle interfacce grafiche, dove il passaggio alla versione 4 di X ha però decisamente migliorato la stabilità del sottosistema grafico.
L'ampia disponibilità di file-system journaled o del sottosistema RAID software mette inoltre, a costi molto accessibili, al riparo da perdite accidentali di dati su disco, rendendo anche una semplice macchina casalinga un sufficiente sistema dalle caratteristiche professionali di base.

Tenetelo lontano da Internet!

Anche gli scenari di uso di Internet dei due sistemi operativi sono radicalmente differenti. Xp rompe una tradizione consolidata da Microsoft di assoluta insicurezza operativa su Internet. Aggiunge sì un cent, ma non molto di più di un cent, di protezione. A seguito delle straordinarie proteste dei clienti, ma soprattutto dei tecnici e delle più importanti società di analisi tecnologica come ad esempio Gartner Group tanto per citare la più importante, Microsoft è stata letteralmente obbligata a rivedere la propria strategia nei confronti della sicurezza dei propri prodotti - casomai avesse mai avuto una strategia del genere. E' stata finalmente trascinata a prendere in considerazione questo tipo di problemi e assumersi una parte delle responsabilità per aver progettato e realizzato sistemi praticamente senza la minima protezione rispetto ad un uso malevolo o anche solo casulaamente errato. La montagna ha però partorito il classico topolino. Anzi la pulce del primo firewall integrato in un S.O. Microsoft. Un tentativo che considerare solo ridicolo è presumibilmente esercitare eccessiva bontà nei confronti del produttore di Seattle. Il firewall di Xp chiude la macchina, se correttamente configurato, da attacchi esterni, ovvero che originandosi da internet provano a prendere il controllo del sistema, ma non si occupa di eventuali connessioni originate dalla macchina stessa verso l'esterno. Questo tipo di "attacchi" sono, in realtà quelli più diffusi e si devono alla miriade di piccoli software o librerie scritte appositamente per fornire i servizi più disparati (e spesso inutili) per cui gli utenti ignari spesso li installano con molta superficialità e nel contempo comunicare all'esterno i dati personali, le abitudini o i percorsi di navigazione del proprietario della macchina. Per non parlare dell'eventuale partecipazione della macchina ad un attacco distribuito DOS (Denial of Service) contro qualche server di rete, senza che neppure il proprietario sappia nulla, che è il principale problema dei client Windows presenti in Internet. Nulla è stato studiato neppure per la protezione dall'annoso problema dei virus. Xp insomma andrebbe usato su Internet solo con un antivirus aggiornato ed attivo e dietro ad un firewall degno di questo nome. Oppure, per tutelare tutti gli altri utenti di Internet, tenerlo ben lontano dalla Rete.
L'invulnerabilità ai virus di sistema e la presenza di un vero firewall nel kernel, rende invece Linux assolutamente sicuro, purchè correttamente configurato e questo, come sempre, è il solo vero problema.

Conclusioni

Sebbene Xp abbia inondato i canali di distribuzione pirata ancor prima che fosse disponibile nei negozi ed è disponibile sulle bancarelle improvvisate dei venditori ambulanti nelle strade principali di tutte le città a poche migliaia di lire a CD-Rom ed è quindi in pratica più facilmente recuperabile di quanto non sia lo stesso Linux, il suo costo di listino è praticamente proibitivo. Anche l'acquisto di un computer nuovo preinstallato con Xp costa non meno di un paio di centinaia di migliaia di lire di differenza che non senza (e almeno adesso si può scegliere di comprarlo senza). Nessuna persona sensata spenderebbe la metà del costo di un moderno PC Entry level per il solo acquisto di un sistema operativo come Xp. Ciò nonostante Xp è veramente la scelta migliore per chi è davvero alle prime armi con il mondo dell'informatica. Posto che voglia pagare salatissimo questo ingresso e che gli vada bene già a priori di non poter muovere poi troppi passi oltre. L'acquisto di un nuovo computer è forse proprio l'evento giusto, invece, per decidere il proprio primo approccio con Linux. Oltre a risparmiare qualche centinaia di migliaia di lire, fatto mai disprezzabile, si avranno le migliori condizioni per apprezzare Linux e per entrare nel mondo dell'informatica con la consapevolezza di costruirsi una vera competenza personale e non rimanere alla mercè dei mutamenti d'umore di un produttore commerciale.

Giacinto Donvito

PfaEdit: una fonte di creatività!

I font sono uno strumento fondamentale per chi voglia fare della computer grafica un'arte. Linux anche questa volta surclassa gli avversari mettendo a disposizione PfaEdit, un tool semplice e potente per la creazione di font.

I font sono una componente molto affascinante del mondo della computer grafica.

Così come molti esperti studiano la grafia umana, e da questa traggono informazioni su personalità, problemi e altro, allo stesso modo un'informazione, elettronica o stampata a partire dal mezzo elettronico, raggiunge persone diverse in modo diverso in funzione della grafica che essa possiede. L'appassionato di computer grafica ha dunque bisogno di poter accedere ad una vasta collezione di font per portare a termine con discreto successo il proprio lavoro.
Nella rete è possibile trovare collezioni di font libere o a pagamento, e addirittura tool per l'amministrazione, installazione e creazione. PfaEdit dà qualcosa in più. Con PfaEdit potremo infatti modificare, convertire e creare da zero intere collezioni di font.

Font di ieri e di oggi

I font vengono alla luce molto dopo la nascita dei primi computer.
I primi computer non erano infatti muniti di monitor, e gli output venivano semplicemente diretti verso una stampante o qualcosa di simile. I successivi progressi tecnologici consentirono agli utenti di usare i primi display e monitor. Ingegneri e informatici si trovarono così a dover risolvere l'annoso problema dei font. I primi terminali video funzionavano proprio come un terminale shell di linux, non avevano la possibilità di sfruttare interfacce grafiche. L'uso del

computer per la creazione di grafica di qualità era indubbiamente ancora fantascienza. I terminali iniziarono però ad usare font che oggi definiamo scalari; si trattava di font molto semplici, obsoleti oggi, ma ancora in uso. Un font di tipo scalare è costruito come un'immagine bitmap; esso può occupare una determinata area, e proprio come in un'immagine bitmap, detta anche scalare, perde di qualità se ingrandito. I font scalari divennero presto obsoleti.
Con l'avvento degli ambienti grafici, (vedi Motif, Mac, etc.), gli utenti iniziarono a sentire il bisogno di una soluzione più performante.

Pose rimedio al problema Adobe, (sempre lei, N.d.R.) con l'invenzione del Postscript:, una soluzione integrata per la stampa e il video. Il Postscript, nato negli anni '70, a conferma della sua qualità ha avuto bisogno nel corso degli anni di poche revisioni, (oggi siamo arrivati al "PostScript Level 3"), ed è considerato ancora oggi come il migliore sistema di stampa.
Esistono molte varietà di font Adobe PostScript. Per esempio possiamo avere i "Font Type 1" o i "Font Type 3", la cui differenza oggi consiste solo nella qualità, inferiore negli ultimi, ma in passato andava a toccare anche problemi di licenze: i *"Font Type 3"*, a differenza dei cugini *"Font Type 1"* erano infatti sviluppati utilizzando un protocollo aperto. I font PostScript hanno il vantaggio di avere un'ottima resa in fase di stampa, specialmente se abbiamo la possibilità di utilizzare stampanti che supportano il PostScript a livello hardware.

Si tratta però di macchine molto costose, dotate di proprio processore e propria RAM, adatte a lavori di un certo livello.
I "Font Type 1", i più diffusi oggi, sono composti da due file fondamentali: i font veri e propri, e le metriche.
I file dei font descrivono la forma astratta del carattere (es. la "o" è una corona circolare, la "l" è una linea verticale, ecc.), e le metriche ne fissano le mutue distanze e le dimensioni. A loro volta i font e le metriche possono assumere formati diversi. I font, identificati con le estensioni **.pfa (Printer Font ASCHII)* o **.pfb (Printer Font Binary)*, si differenziano fra loro nel sistema di descrizione: ascii o binario rispettivamente; mentre le metriche saranno disponibili nelle estensioni **.pfm (Printed Font Metrics)* e **.afm (Adobe Font Metrics)*.

Figura 2

La mappa dei caratteri.

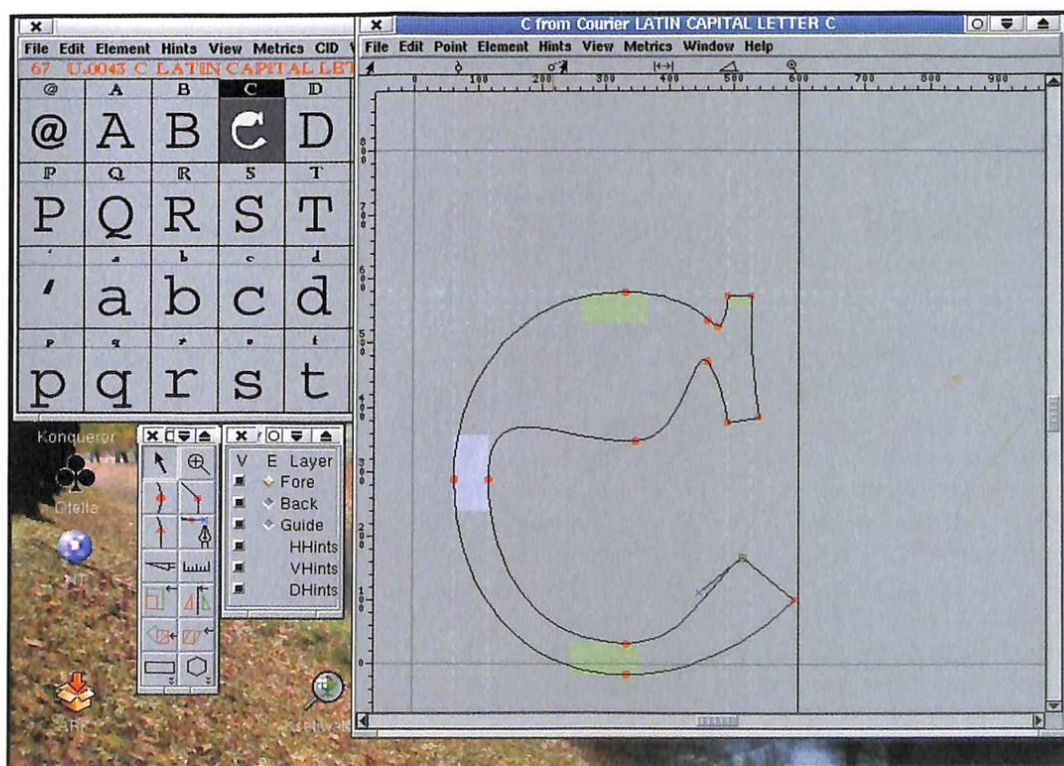



Figura 1

Definizione di un carattere.

...e Linux?

I crescenti sistemi operativi dell'epoca non

accoglievano con calore la proposta di Adobe, in quanto nessuno era propenso a subordinare una parte così importante del sistema operativo a terzi. Gli interessati trovarono quindi un compromesso nel *True Type Font:* un nuovo font semplice, libero, e che fosse portabile su più piattaforme. I font **.ttf* hanno forse una resa leggermente minore dei font proposti da Adobe, ma sono allo stesso tempo più semplici da gestire, in quanto tutte le informazioni, (font e metrica), sono contenute in un unico file. La diffusione dei TTF è stata capillare, tanto da rendere questo formato uno "standard de facto". Linux supporta, nelle ultime versioni di XFree 4.1 (il server video dei sistemi GNU-Linux), una gran varietà di tipologie di font e fra queste anche i font TTF e PostScript Type 1 e 3. I suddetti font, grazie ai grandi miglioramenti apportati a CUPS e altri server di stampa liberi, possono anche essere stampati con fantastici risultati. Nella rete è infine possibile trovare una gran quantità di font gratuiti molto spesso distribuiti nel formato **.ttf;* se le collezioni preformate non ci aggradano possiamo oggi creare la nostra collezione privata utilizzando un ottimo sistema di sviluppo dei font visuale, pratico, veloce e intuitivo: *PfaEdit.*

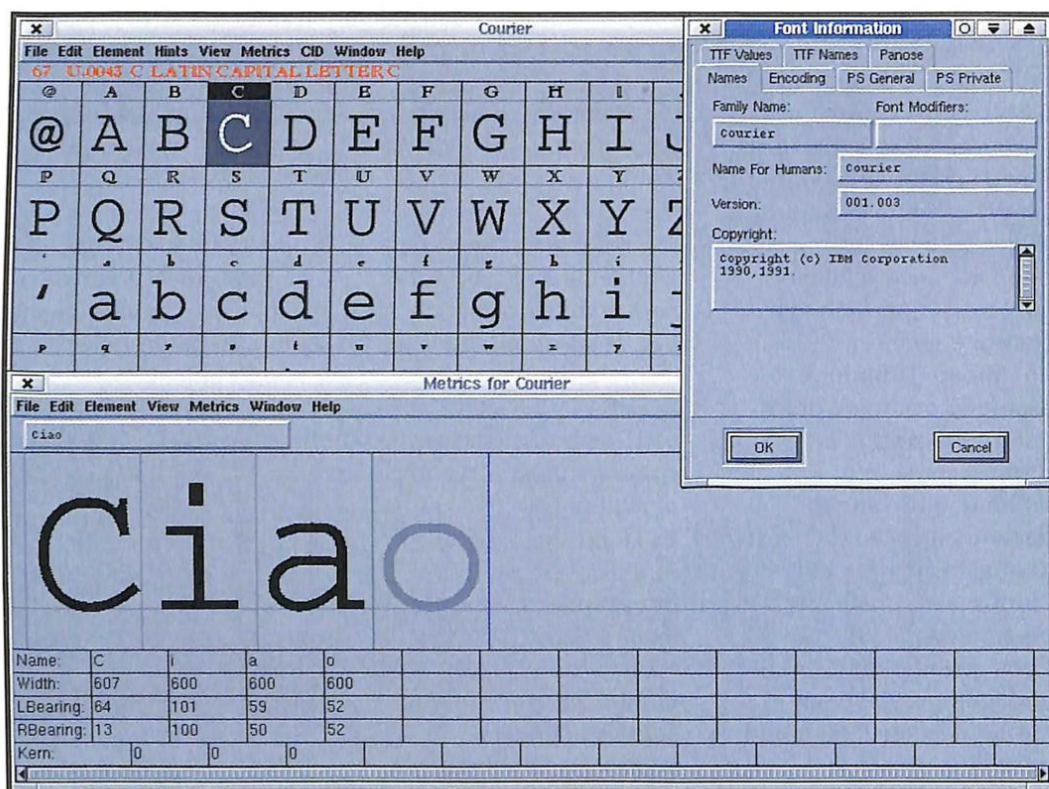



Figura 4

Parametri relativi al font Courier.

PfaEdit, per iniziare

PfaEdit è stato creato da George Williams e successivamente reso open source e distribuito con licenza GPL. L'interfaccia è spartana, ma pratica e veloce. Le prove sono state effettuate utilizzando PfaEdit sia su una RedHat 7.2 che su una Mandrake 8.1. In entrambi i casi affidabilità, velocità e semplicità di installazione sono state perfette. *PfaEdit* è disponibile sulle pagine ufficiali del sito, *(http://pfaedit.sourceforge.net),* in formato sorgente *(*.tar.gz),* o binario precompilato. Sono al momento disponibili pacchetti precompilati *RPM, DEB* e *TGZ* per piattaforme *i386, SUN Solaris, Linux Power PC,* e recentemente anche *MacOS-X.* Il sito ufficiale di PfaEdit è realizzato molto bene: bello, moderno e completo. Anche la documentazione a corredo è valida e chiara. Purtroppo sito e documentazione sono disponibili solo in lingua inglese.

Cosa può fare

Modificare o creare un font con PfaEdit è un gioco da ragazzi. Basterà avviare PfaEdit e selezionare il font da modificare scorrendo le directory o optare per un nuovo font. Gli sviluppatori di PfaEdit hanno creato anche un particolare formato per salvare i font da noi creati, lo **.sfd* acronimo di *Sample Font Database,* che ci verrà proposto ad ogni salvataggio. Modificando un font esistente vedremo apparire una grande tabella con un menù a tendina nella parte alta. Se i caratteri sono rappresentati in scala troppo piccola, potremo selezionare il menù a tendina *View* per

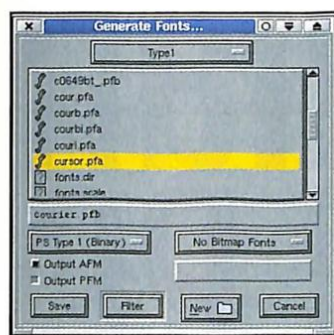



Figura 5

Maschera per la generazione di un font.

modificarne la outline. PfaEdit permette inoltre di attivare l'Anti Alias, una tecnica di interpolazione che permette di rappresentare i font con un contorno più morbido. La feature si trova sempre nel menù a tendina *View*.

Attiviamola per notare la differenza. Cliccando adesso su un qualsiasi carattere vedremo aprirsi una nuova finestra con all'interno la costruzione geometrica del font. Come già detto i font TTF e PostScript sono vettoriali, ciò significa che sono descritti da regole matematiche che ne consentono l'ingrandimento e il rimpicciolimento senza apparente perdita di qualità. Ma torniamo a PfaEdit. Noteremo sicuramente la presenza di due finestre fluttuanti. La prima destinata ad ospitare gli strumenti per la creazione/modifica del font consente di inserire nodi di varia natura, scalare, piegare, tagliare, ecc. La seconda contiene al suo interno degli switch che permettono l'accensione e lo spegnimento di alcune parti del carattere come il contorno, le linee guida delle metriche.
Costruito o modificato il nuovo font, dovremo modificare le impostazioni relative alle metriche, cioè le impostazioni relative alle mutue distanze fra i caratteri. Dal menù a tendina basterà selezionare *File --> Open Metrics,* per vedere comparire una nuova finestra.
La finestra per la modifica delle metriche è quanto di più comodo ci si potesse attendere. La parte alta è caratterizzata da una casella in cui inserire il testo di prova, in seguito, cliccando sui vari caratteri potremo spostare le linee che li contornano e che in un certo senso indicano lo spazio occupato da ogni carattere.
Una volta finito non dovremo far altro che salvare il lavoro fatto e optare per un formato di esportazione.
PfaEdit consente di esportare font nel formato *True Type, Adobe PostScript Type 0, 1* e *3,* utilizzando una qualsiasi combinazione di font e metrica (**.pfa* e **.pfb* per i font, **.afm* e **.pfm* per i file metrica) e anche nei formati bitmap a *75* e *100* punti.

Figura 6

La definizione di un carattere.

Conclusioni

PfaEdit, disponibile all'indirizzo *http://pfaedit.sourceforge.net,* è un software potente, maturo e completo. Il sito Internet è parimenti semplice ma completo e la documentazione a corredo è più che buona. Particolare merito a PfaEdit inoltre nella cura della costruzione della metrica, molto delicata, che è in questo caso, molto ben implementata. Purtroppo nei font gratuiti reperibili in rete questa parte non è di solito molto curata. Una buona occasione per iniziare ad usare questo fantastico software.

Andrea Cavaleri

GNOKII, cellulari a spasso con il PC

Qualsiasi "smanettone" degno di questo nome sarà stato tentato ed incuriosito dalla possibilità di collegare un telefonino GSM al proprio computer, soprattutto se si tratta di un notebook, in modo da realizzare un sistema realmente portatile per il collegamento ad Internet, o comunque per scambiare informazioni.

Il collegamento cellulare-computer ha anche altri vantaggi, non strettamente legati alla comunicazione. Infatti, quasi tutti i modelli di telefono cellulare oggi in commercio, sono molto di più di strumenti per la comunicazione vocale, sono delle agende, delle rubriche telefoniche ed altro ancora. Interessante sarebbe quindi poter utilizzare queste funzionalità per mezzo di un computer, basti pensare alla possibilità di gestire la rubrica telefonica, o di automatizzare l'invio e la ricezione degli SMS.
Ma non facciamoci prendere la mano...
Se nel mondo Microsoft le cose sono abbastanza chiare e documentate, non altrettanto possiamo dire per quello del nostro caro pinguino. Infatti come avviene per altre componenti e periferiche, non sempre le case produttrici di hardware sono disposte a divulgare le specifiche per permettere agli sviluppatori open source di scrivere applicazioni o driver analoghi a quelli "ufficiali", anche se questa tendenza tende piano piano a scomparire.
In questo articolo vedremo cosa è attualmente possibile fare con *Gnokii*, il clone del *Nokia Data Suite,* l'applicativo distribuito dalla casa finlandese per il collegamento al PC di alcuni modelli dei suoi telefoni cellulari.

Le diverse soluzioni di collegamento

Oggi esistono essenzialmente 2 soluzioni per collegare un telefonino cellulare al PC: tramite collegamento seriale o tramite interfaccia ad infrarossi. Entrambe le soluzioni presentano vantaggi e svantaggi. Per utilizzare il collegamento seriale è necessario dotarsi dell'apposito cavo di collegamento, questa rappresenta senza dubbio la soluzione più tradizionale ed assicura la compatibilità praticamente con qualsiasi PC. Mentre il collegamento tramite raggi infrarossi è utilizzabile solo sui telefonini dotati dell'apposita interfaccia e comunque ha dei campi di utilizzo più limitati a causa dei vincoli imposti dai raggi infrarossi, come ad esempio la distanza tra i dispositivi ed il loro allineamento. D'altra parte anche questa soluzione presenta alcuni vantaggi, come la possibilità di poter collegare il cellulare a dispositivi non dotati di seriali standard, come ad esempio i palmari, e di non necessitare di cavi aggiuntivi.

Gnokii

Per capire il ruolo svolto da Gnokii, nel collegamento tra telefonino cellulare e computer, bisogna distinguere tra i diversi sistemi di co-

municazione dati implementati nei telefonini GSM. Leggendo le specifiche tecniche di alcuni cellulari può capitare di imbattersi nella voce *"modem integrato"*, in realtà il termine *modem* abbinato ad un telefonino GSM è alquanto improria. Più correttamente con questo termine ci si riferisce ad un modulatore/demodulatore di segnali, cioè ad un dispositivo in grado di effettuare una conversione dei segnali da analogico a digitale e viceversa. Conversione che si rende necessaria, ad esempio, quando utilizziamo un canale di comunicazione analogico, come la comune linea telefonica, per collegarci ad Internet o comunque scambiare dati con un altro computer. Allora che genere di modem intendono? Generalmente con la dicitura *"modem integrato"* viene indicata la possibilità di utilizzare il cellulare come mezzo per lo scambio di dati digitali tra computer. Se ci pensiamo bene i telefoni GSM sono per loro natura digitali, così come digitale è il formato dei dati inviati dal computer. Il modem all'interno di questi GSM non è altro che un firmware che permette al telefonino di essere visto dal nostro PC come un qualsiasi modem Hayes compatibile. Utilizzando un cellulare con queste caratteristiche basterebbe collegarlo al nostro PC ed utilizzarlo come un comune modem. Purtroppo, non è sempre così semplice... Infatti alcune case costruttrici utilizzano una filosofia diversa, lasciando al cellulare il solo compito di trasmettere e ricevere segnali e demandando ad un applicativo (chiamato anche *softmodem*) il compito di simulare l'interfaccia di comunicazione *Hayes*. Questa soluzione ha il vantaggio di poter essere migliorata o comunque modificata intervenendo solo sul software, ma ha lo svantaggio di richiedere l'acquisto dell'applicativo di gestione e non è detto che questo sia disponibile per il nostro sistema operativo! Tra le case che adottano questo sistema troviamo la *Nokia*, con il software applicativo *Nokia Data Suite (NDS)*, giunto alla versione 3.0 e disponibile per sistemi operativi Microsoft. *Gnokii* rappresenta l'alternativa open source all'ufficiale NDS, quindi ci permetterà di collegare i telefonini Nokia al nostro linux box. Purtroppo non tutti i modelli di cellulare prodotti dalla Nokia sono utilizzabili con Gnokii, ma la lista dei telefonini utilizzabili con questo applicativo cresce di mese in mese. Per i modelli attualmente supportati fare riferimento alla Tabella 1. Il supporto per altri modelli più recenti ed utilizzati, come il 3210, 3310 o 7110, è in fase si sviluppo. Per la nostra prova utilizzeremo un collegamento seriale, realizzato tramite cavo FBUS, con un Nokia 6110 su una distribuzione Slackware 8.0, ma naturalmente sarà possibile utilizzare Gnokii anche con la vostra distribuzione preferita.

6110	6130	6150	6190	5110	5130	5190
3110	3810	8110	8110i			
2110	2140	6080				
640						
5160	6160	6185				

Tabella 1

Modelli di telefonini Nokia che sono supportati da Gnokii.

Installazione

Passiamo ora dalla teoria alla pratica realizzando insieme tutti i passaggi necessari per installare Gnokii sul nostro sistema linux. Naturalmente, per prima cosa, dobbiamo prelevare il software dal sito ufficiale *http://www.gnokii.org* (Fig. 1), cliccando sulla sezione *download*. Al momento in cui scrivo è disponibile la versione *0.3.3* (stabile), sia in formato sorgente che in binario per le distribuzioni linux più diffuse. Per la nostra prova utilizzeremo i sorgenti, che provvederemo a configurare e compilare. Una volta scaricato il pacchetto dovremo decomprimerlo con il comando:

```
tar -zxf gnokii-0.3.3.tar.gz
```

verrà così creata la directory *gnokii-0.3.3* contenente i sorgenti, la documentazione e tutto il necessario. Il processo di compilazione ed installazione è abbastanza classico, il primo passo è quello di lanciare il configure, eventualmente specificando qualche opzione per meglio adattare Gnokii al nostro sistema. Per avere una lista di tutte le opzioni disponibili digitare:

```
./configure --help
```

Le più utili sono:

```
--prefix=<path di
          installazione>
```

che ci permetterà di specificare un path di installazione diverso da quello di default */usr/local*

```
--without-x
```

per compilare Gnokii senza

Figura 1

Homepage del progetto Gnokii.

le applicazioni per l'interfaccia grafica X-Window. Dopo aver completato con successo la fase di configurazione, possiamo passare alla compilazione vera e propria, per fare questo basterà digitare:

```
./make
```

e se anche in questa fase non abbiamo riscontrato problemi continuiamo con l'installazione vera e propria:

```
./make install
```

Una volta installati i programmi, non ci rimane che ultimare la fase di configurazione in base al nostro hardware. Per fare questo dovremo copiare il file:

```
./Docs/sample/gnokiirc
```

in */etc/gnokiirc* oppure in *.gnokiirc* nella nostra home directory e quindi adattarlo al nostro hardware. In particolare bisognerà indicare il tipo di cellulare che intendiamo utilizzare e la porta sulla quale si trova collegato, assicurandoci di avere i permessi di lettura/ scrittura sulla seriale indicata. Tutto qui!
A questo punto siamo pronti ad utilizzare Gnokii!

Gnokiamo !

Completata la fase di installazione e configurazione siamo pronti a sperimentare il nostro nuovo programma. Per prima cosa dobbiamo assicurarci che la comunicazione tra telefonino e computer avvenga correttamente, per fare questo usiamo:

```
Tera Term - 192.168.1.18 VT
File Edit Setup Control Window Help
root@powerlab:/# gnokii --identify
IMEI:       490518109266443
Manufacturer: Nokia
Model:      NSE-3
Revision: SW 4.33, HW4000

root@powerlab:/# gnokii --monitor
Entering monitor mode...
RFLevel: 4
Battery: 4
Power Source: battery
SIM: Used 13, Free 117
Phone: Used 26, Free 24
FD: Used 0, Free 5
ON: Used 0, Free 2
SMS Messages: UnRead 0, Number 10
Network: Wind Telecomunicazioni SpA (Italy), LAC: a80d, CellID: 7598
Leaving monitor mode...
root@powerlab:/#
```

Figura 2

Test per il corretto funzionamento di Gnokii.

```
gnokii –identify
```

oppure:

```
gnokii –monitor
```

se tutto funziona correttamente il sistema ci fornirà informazioni sul tipo di cellulare rilevato (Fig. 2).
La suite Gnokii è composta essenzialmente da 3 programmi:

gnokii
xgnokii
gnokiid

gnokii ci permetterà di gestire, tramite linea di comando, praticamente qualsiasi aspetto del nostro telefonino Nokia. Potremo leggere e scrivere dati nella rubrica telefonica, leggere ed inviare SMS, leggere e scrivere dati nel calendario oppure impostare data, ora, loghi e suonerie. Per visualizzare la lista delle opzioni disponibili digitiamo:

```
gnokii –help | less
```

Ad esempio, per leggere la prima locazione di memoria della rubrica memorizzata nella SIM useremo:

```
gnokii –getmemory SM 1
```

per leggere gli SMS ricevuti e memorizzati nella SIM, da posizione 1 alla 5, useremo (Fig. 3):

```
gnokii –getsms SM 1 5
```

dove SM indica il tipo di memoria che intendiamo utilizzare.

ME	Memoria interna del telefono.
SM	Memoria della scheda SIM.
MT	Memoria combinata scheda + telefono.
DC	Numeri telefonici delle chiamate effettuate.
RC	Numeri telefonici delle chiamate ricevute.

Tabella 2

Tipi di memoria disponibili. Alcuni di essi sono applicabili ad altre opzioni di gnokii.

```
Tera Term - 192.168.1.18 VT
File Edit Setup Control Window Help
root@powerlab:~# gnokii --getsms SM 1 5
1. Inbox Message (not read)
Date/time: 14/12/1 13:51:56 +0200
Sender: 447166714 Msg Center: +491722270041
Text:
id:d3dbdee5

SMS ricevuto con successo da Gnokii...


SMS location SM 2 empty.
SMS location SM 3 empty.
SMS location SM 4 empty.
SMS location SM 5 empty.
root@powerlab:~#
```

Figura 3

Lettura degli SMS tramite linea di comando.

Coloro i quali non dovessero trovarsi a proprio agio con i tool a linea di comando o che preferissero le interfacce grafiche, potranno utilizzare xgnokii, l'applicativo che ci permetterà di controllare il nostro cellulare tramite l'ambiente grafico X-Window. Il programma si presenta con un'interfaccia gradevole ed essenziale (Fig. 4), ed è dotato di

Figura 4

L'interfaccia di xgnokii.

alcune funzionalità aggiuntive rispetto a gnokii, che ci consentiranno di avere un ambiente grafico completo per la gestione del nostro telefonino.
Tra queste troviamo:

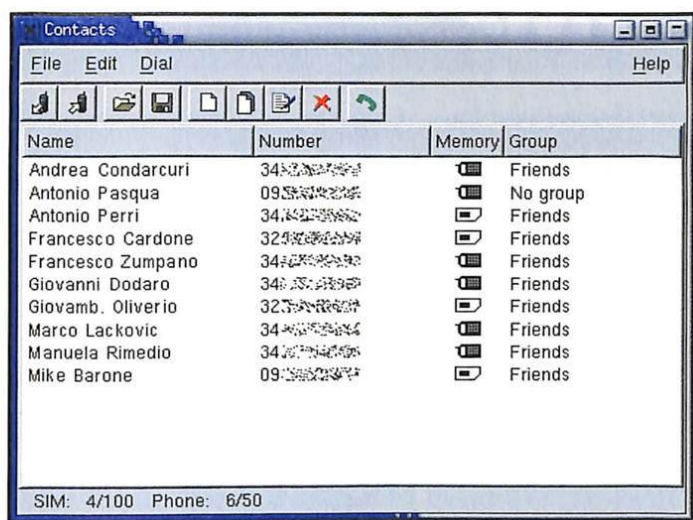



Figura 5

Gestione della rubrica telefonica.

contacts: permette di gestire, in ogni suo aspetto, la rubrica telefonica del cellulare, sia quella contenuta nella scheda SIM che quella contenuta nella memoria del telefonino (Fig. 5). Oltre a consultare, inserire e modificare i dati, sarà possibile scaricare interamente la rubrica sul nostro hard-disk, in modo da conservarne una copia di backup, oppure compiere l'operazione inversa, cioè sincronizzare la rubrica del cellulare con quella memorizzata sul disco.

sms: si presenta come un client di posta elettronica e ci permette di leggere e cancellare gli SMS presenti nel telefonino. Inoltre consente di scrivere nuovi SMS, inviarli oppure memorizzarli nel telefonino per spedirli successivamente.

calendar: ci permette di gestire il calendario, con tanto di agenda, del nostro cellulare. Sarà possibile inserire, modificare e cancellare piccole annotazioni, che ci aiuteranno a ricordare i nostri appuntamenti.

logos: questo è un piccolo editor grafico che ci permette di personalizzare i vari logo disponibili sui telefoni Nokia e precisamente: di accensione, di operatore e di gruppo. Su internet si trovano moltissimi siti da cui scaricare i logo più svariati, con questo editor sarà possibile eventualmente modificarli per poi uploadarli sul telefonino. Comunque niente ci impedisce di crearne di nostri, e con un po' di impegno potremo ottenere simpatici risultati (Fig. 6). Completamente diverso è invece il compito svolto da *gnokiid*. Questo è il componente che si occupa dell'emulazione del modem, in pratica implementa il softmodem. *Gnokiid*, una volta lanciato, creerà un dispositivo virtuale *(virtual modem)* identificato dal link */dev/gnokii*. Questo potrà essere utilizzato da tutte le altre applicazioni come un qualsiasi modem tradizionale, permettendoci di effettuare chiamate dati verso altri computer e collegamenti PPP. In aggiunta ai classici comandi AT, sarà possibile utilizzare alcune estensioni appositamente realizzate per i telefoni cellulari e conosciute con il nome di comandi *AT+*. La funzione principale di questi comandi aggiuntivi è quella di permetterci l'accesso alle funzioni tipiche dei cellulari, come la rubrica o gli SMS (Fig. 7). Tramite *gnokiid*, i comandi *AT+* ed un qualsiasi linguaggio di programmazione che permetta la comunicazione con dispositivi seriali, possiamo scrivere senza troppe difficoltà dei programmi capaci di leggere e manipolare i dati contenuti nel nostro telefonino. Basti pensare alle sole applicazioni che utilizzano gli SMS come sistema di comunicazione o di notifica di eventi.

Figura 6

Creazione di un logo tramite xgnokii.

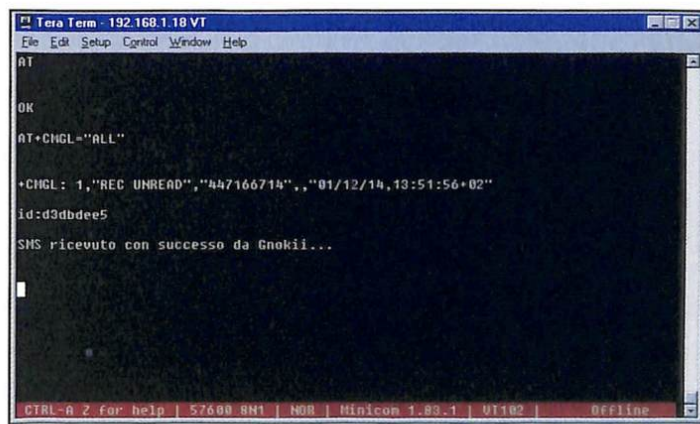
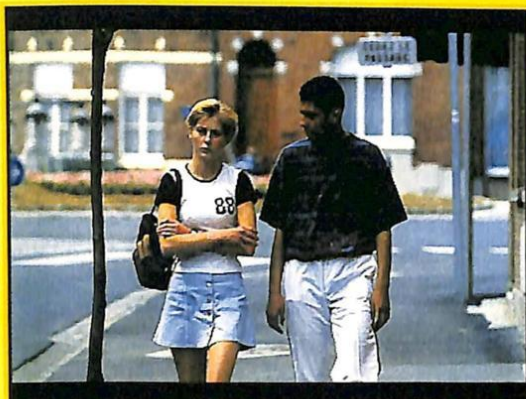
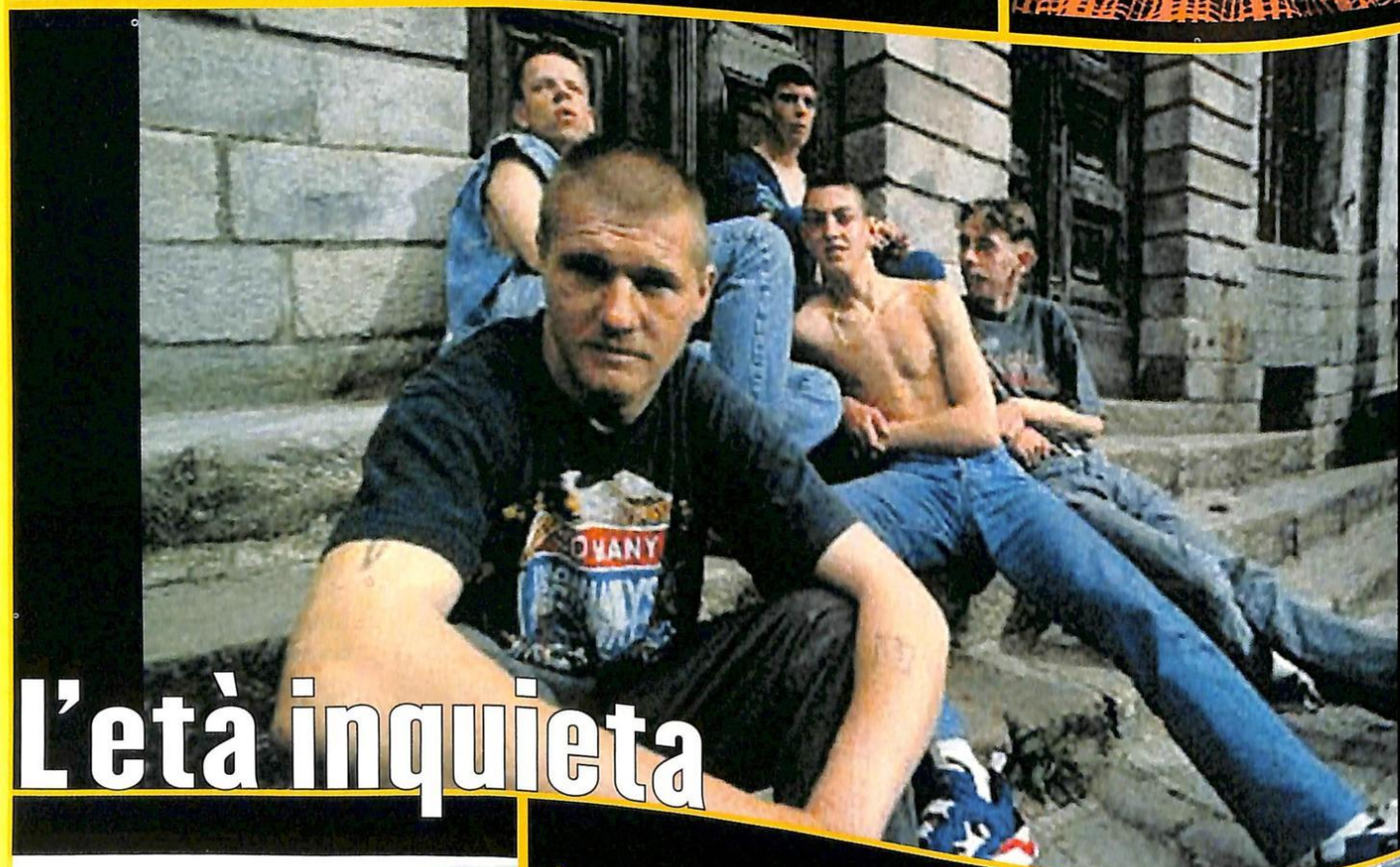
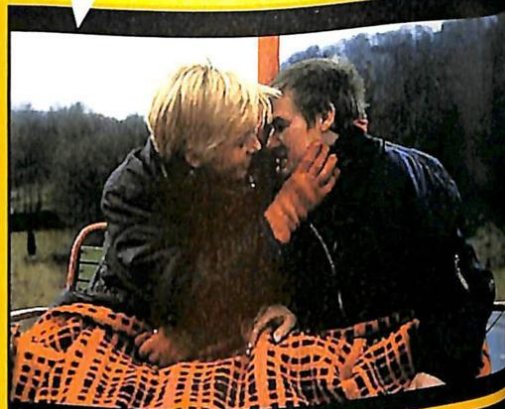
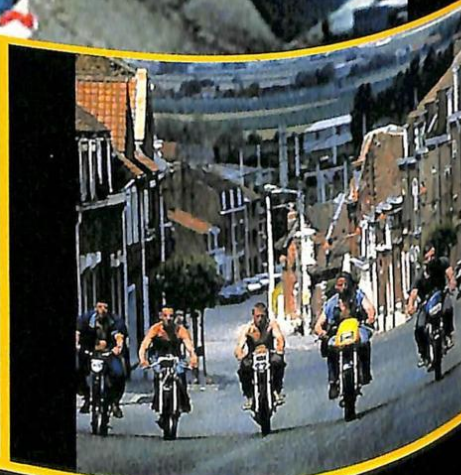



Figura 7

Esempio di utilizzo dei comandi AT+.

Conclusioni

Abbiamo visto come sia possibile controllare e gestire, tramite Gnokii, le funzionalità di un telefono cellulare. Certo ancora c'è del lavoro da fare, in primo luogo quello di estendere il supporto ad altri telefonini, ma certamente abbiamo avuto un'idea delle interessanti applicazioni che è possibile realizzare.

Antonio Pasqua

Francia 1997

Durata
96 min

Regia
Bruno Dumont

Fotografia
Philippe Van Leeuw

Le distribuzioni di nuova generazione (RedHat, Suse, Mandrake) necessitano di harware sempre piu' potente? Una continua rincorsa ad upgrade per poter godere degli ultimi tool grafici? Possibile che nessuno abbia mai pensato a creare mini distribuzioni linux mirate a sfruttare hardware obsoleti? Magari utilizzando i vecchi (mai dimenticati) floppy disk? Non temete: continuate la lettura e vedrete...

Rottamazione? No Grazie!

Quanti PC 486 abbiamo rottamato?Abbandonato nelle nostre soffitte? Possibile che non esista alternativa? Il mio Prof. di Statistica amava dire: "La necessita' e' madre di ogni invenzione"...con il senno di poi...beh...aveva ragione.

Un pizzico di MBR, una manciata di byte...

Le informazioni sul partizionamento dell'hard disk sono contenute all'interno della prima traccia del primo settore del disco *(Master Boot Record)*. Ad ogni avvio, il BIOS della scheda madre ne leggerà i dati, inizializzando questo settore *(MBR)*. All'interno del *Master Boot Record* è contenuto un piccolo programma che, ad ogni avvio, leggerà la tabella delle partizioni e controllerà l'esistenza di una partizione attiva leggendone il primo settore. In questo settore è contenuto il *'Boot Sector'* che si occuperà di caricare il sistema operativo (controllo dell'effettiva bootabilità del sistema).

Approssimativamente il settore contenente l'*MBR* ha una dimensione di 512 byte e sarà fondamentale per permettere a loader successivi di riconoscere il tipo di partizionamento e caricare il sistema operativo.

Solitamente le partizioni possono svolgere funzione di: (rif. *BootDisk Howto*)

- *boot;*
- *root;*
- *boot/root;*
- *utility;*

Una partizione *'boot'* contiente sia l'immagine del kernel sia il *boot loader*, mentre una partizione *'root'* contiene un filesystem e il necessario per operare in ambiente Linux (la possibilità di avere una partizione *'root'* non correlata con le altre, consente all'utente di poter operare in modo indipendente, nessun bisogno di correlarsi a partizioni differenti). Comprensibile intuire che spesso la partizione *'utility'* viene contenuta all'interno della partizione *root* (differentemente si parlerà di partizione singola contenente programmi utili all'utente). La partizione *'root'* non deve necessariamente contenere l'immagine del kernel utilizzata dal sistema: una volta inizializzata la procedura di loader, il sistema sarà comunque in grado di interagire. Gli attributi che gli permettono questo sono:

- File system;
- set di directory *(dev, proc, bin, etc, lib, usr, tmp);*
- set base utility

(bash,ls..);

- set file di configurazione *(rc, inittab, fstab..);*

- set base librerie.

Come posso, viste le ovvie restrizioni di spazio del floppy disk, inserire tutte queste feature all'interno del piccolo supporto magnetico? Non si potrebbe montare il file system in memoria partendo da un'immagine compressa? Come fare? Gli sviluppatori del kernel hanno dato risposta al nostro problema con una funzione: *loop device*. Abilitando la voce *'Loopback device support'* (la si puo trovare nella sezione *'Block device'*) sarà consentito usare un singolo file come *block device*. È possibile quindi, creare un file system nel *block device* e utilizzarlo come *mount point*.
L'identificativo del device nella directory *dev* è:

```
/dev/loopN ('N' numero
      indicativo device
           utilizzato)
```

Prima di addentrarci nella creazione vera e propria della nostra distribuzione tascabile è utile sapere che il supporto magnetico floppy disk può avere formattazione variabile tra i 1.44Mb a 1.99Mb. Vista la penuria di spazio e la molteplicità di funzioni che, ad oggi, sono richieste ad una distribuzione baby, consiglio di utilizzare la formattazione a 1.68Mb (scelta diplomatica vista l'instabilità/compatibilità della formattazione 1.99Mb). All'interno della distribuzione inseriremo:

- filesystem tipo ext2;

- kernel della famiglia 2.2.x;

- supporto ADSL (alternativamente ISDN);

- SSH.

I requisiti necessari per portare a buon fine il lavoro sono:

- un floppy ad alta densità;

- un loader (syslinux);

- un sistema linux dal quale fare un grab delle parti necessarie;

- un'immagine del filesystem root.

Iniziamo...

Preliminarmente occorre formattare il nostro floppy disk. Da shell digitiamo:

```
fdformat /dev/fd0u1680
(device per supporti 1680)
```

seguito da:

```
mkfs -t (tipo di
    filesystem da creare)
    msdos /dev/fd0u1680
                   1680
```

Indispensabile scegliere il loader. Solitamente le distribuzioni su floppy si avvalgono del loader *'syslinux'*: scelta consigliata vista il minor spazio occupato su floppy e la maggiore usabilità di questo. Scaricare *'syslinux'* all'url: *http://freshmeat.net/redir/syslinux/10177/url_bz2/syslinux-1.63.tar.bz2* (ultima release la 1.63). *Syslinux* ci permettere, partendo da un filesystem di tipo *FAT*, di inizializzare un kernel linux. Per creare il *boot loader* in ambiente linux digitiamo:

```
syslinux /dev/fd0 (in
 ambiente DOS basterà di-
      gitare 'syslinux'
    seguito dal nome del
             device -a:-).
```

Il programma genererà all'interno del dischetto il file *'ldsyslinux.sys'* e scriverà un settore del disco al quale sarà delegata l'inizializzazione del file.

Figura 1

Requisiti necessari alla costruzione di una baby distribuzione.

Non rimane che inserire nella directory *radice*, un'immagine del kernel (convenzionalmente denominata *'linux'*). La possibilità di creare un file di configurazione per gestire il boot loader ci permette di adattare alle nostre esigenze la baby distribuzione. Il file avrà nome *'syslinux.cfg'* e rappresenta un normale file di testo (generabile da qualsiasi editor): il file gode di una struttura simile a quella di *'lilo.conf'* (potete controllarne la sintassi editando il file *'/etc/lilo.conf'*) con l'eccezione che la correlazione tra kernel, settore d'avvio e configurazione viene gestita da *'ldlinux.sys'*. Un file di configurazione tipico potrebbe essere (*Appunti di Informatica Libera*):

```
DEFAULT linux
TIMEOUT 0
DISPLAY INTRO.TXT
PROMPT 1
F1 INTRO.TXT
F2 VARIE.TXT
LABEL linux
  KERNEL LINUX
LABEL floppy
  KERNEL LINUX
  APPEND " initrd=
           initrd.gz root=
           /dev/fd0u1680"
```

Dove:

- *DEFAULT linux*: viene specificato il file immagine da utilizzare;

- *TIMEOUT 50*: valore da impostare (50 è pari a 5 secondi) prima che il boot loader in automatico inizializzi il sistema (ponendo il valore pari a *0* l'utente dovrà necessariamente interagire con la macchina);

- *DISPLAY INTRO.TXT:* file contenente le informazioni che l'utente visualizzerà in console in fase di boot;
- *PROMPT 1:* parametro che consente all'utente di inserire comandi se il campo avesse valore pari a *0* il boot loader non interrogherà l'utente);
- *F1 INTRO.TXT:* visualizzazione del file *'intro.txt'* attraverso la pressione del tasto funzione 2;
- *F2 VARIE.TXT:* visualizzazione del file di testo *'varie.txt'* attraverso la pressione del tasto funzione 2;
- *LABEL linux:* definizione dell'etichetta;
- *KERNEL LINUX:* indica di utilizzare il file *'linux'* come immagine del kernel quando viene selezionata l'etichetta *'linux'*;
- *APPEND "initrd= initrd.gz root= /dev /fd0u1680":* stringa necessaria per permettere al boot loader di caricare l'immagine compattata dal device *fdou1680*.

Nell'append finale è stato espresso il dispositivo floppy per il mount del filesystem. Altri attributi utilizzabili nella definizione sono:

- *ro:* definizione di accesso al file system in sola lettura;
- *rw:* definizione di accesso al file system in lettura e scrittura (read-write);
- *mem:* possibilità di definire la dimensione di memoria ram (espressa sia in esadecimale che in unità decimali);
- *init:* definizione del path del programma che deve svolgere funzione di *'init'* (solitamente il kernel cera autonomamente il file *'/sbin/init'* o, alternativamente, il file *'/etc/init'* -insolito-);

è utile ricorrere all'attributo *'display'* per permettere ad eventuali utenti di interagire con il boot loader: la possibilità di ricorrere all'help tramite tasti funzione non viene esplicitamente espressa in fase di boot.

FileSystem

Risulta fondamentale disporre di una distribuzione linux funzionante per creare/simulare la nostra baby distribuzione. Creata una directory ad hoc per contenere il sistema *(mkdir /bkane)* assicuriamoci di aver compilato il kernel con il supporto per:

- device loop;
- ram disk;
- initrd.

La necessità di possedere il *'device loop'* è già stata descritta nella prima parte dell'articolo. Perché abilitare *'ram disk'* e *'initrd'*? Semplice:

- *ram disk:* abilitando questa funzione sarà possibile utilizzare una parte della nostra ram come block device permettendoci di compiere tutte le operazioni di lettura/scrittura tipiche dei device, a blocchi *(ex:hard disk)*
- *initrd: 'initial ramdisk'* è un ram disk che è caricata dal boot loader (*syslinux* nel nostro caso) e che viene montata come *'root'* prima della normale procedura di boot.

Come montare un file system in loop? Da linea di comando digitiamo:

```
mount -t ext2 -o loop
      bkane.img /loopdev
```

Controlliamo l'effettivo mount della nostra immagine utilizzando il comando *'df'* (nella tabella di mounting dovrà essere presente la nostra immagine). Entriamo nella directory *'/bkane'* e, utilizzando il comando *'dd'*, generiamo il file che conterrà il nostro file system (l'utility *'dd'* servirà per copiare un file dallo standard *input* allo standard *output*).
Da shell digitiamo:

```
dd if=/dev/zero of=
  initrd bs=1k count=3000
```

Dove con:

- */dev/zero:* rappresentazione dello standard *input* (device contenente solo *0*);
- *initrd:* rappresentazione dello standard *output;*
- *bs:* espressione di quanti *k* verranno scritti ogni volta;
- count: numero di scritture.

Il file di output generato *(initrd)* conterrà solo numeri zero: questo ci consentirà di avere uno spazio, non occupato da utility, un valore costante. Creato il file non rimane che generare un filesystem al suo interno: utilizziamo il comando *mke2fs*. Da shell digitiamo:

```
mke2fs -m 0 -i 2500
             initrd
```

In questo modo il filesystem creato disporrà di *0* byte riservati all'utente root e 2500 byte per ogni inode. Generiamo una directory *'linux'* (controllare il file di configurazione di *syslinux* riportato ad inizio articolo) e montiamolo:

```
mkdir /bkane/linux
mount -t ext2 -o loop
      initrd /bkane/linux
```

Fatto questo (accertarsi dell'effettivo mount del file system utilizzando il comando *'df'*) non rimane che inserire le opportune directory. Cosa inserire? Cosa ritenere importante? Beh, controllando le radici di quasi tutte le distribuzioni, le directory che trovermo sono:

- */*
- */dev*
- */lib*
- */bin*
- */sbin*
- */usr*
- */tmp*
- */proc*
- */floppy* (possibile creare un directory per il device *cdrom, /cdrom*)

La directory *'device' (/dev/)* conterrà le specifiche delle periferiche che noi vorremo usare. Controllando all'intero di */dev/* della nostra distribuzione fissa possiamo notare quanti siano i */dev/* costruiti di default dal sistema. Si renderà necessaria una scelta. Al fine di mantenere a livello accettabile l'usabilità del baby sistema dovremmo creare i device: console, *cua0-1, fd0, hda1-4* (qualora la vostra macchina non disponga di nessun disco fisso si può evitare di creare questo device), *null, ram 1-4, tty* (necessario ad emettere il nome corrispettivo alla console che si sta utilizzando) *1-3, zero, systty* (definizione le caratteristiche della connessione del terminale). Utilizzando il comando *'mknod'* sarà possibile generare un dispositivo a caratteri o un dispositivo a blocchi (questi dovranno essere contraddistinti sia da un numero primario che da un numero secondario).

Figura 2

Schema delle directory che conterra' la nostra distribuzione.

Alternativamente è possibile ricreare i device copiando quelli già esistenti all'interno della nostra distribuzione fissa. Utilizzando il comando (inserire i device creati/copiati all'intero della directory *'/dev/'*, generata all'interno del direttorio contenente la nostra baby distribuzione):

```
cp -dpR /dev/device
```

Dove:

- *cp:* è indicativo del comando copia;
- *-dpR:* l'attributo *'d'* avverte di copiare link simbolici, l'attributo *'p'* permette di preservare gli attributi del file che si sta copiando e *'R'* permette di copiare le directory ricorsivamente.

Occupiamoci ora della directory *'/lib/'*. All'interno di questa si trovano le librerie utilizzate dal sistema. Dopo aver generato la directory *'/lib'* (vi ricordo che stiamo tentando di simulare in tutto una distribuzione vera e propria) dobbiamo adoperarci a ricopiare al suo interno librerie (corredate da rispettivi link simbolici) sulle quali si basa la nostra distribuzione. Le librerie da copiare saranno (possono variare da distribuzione a distribuzione, utilizzare il comando *'ldd'* seguito dal nome del programma che ci interessa per considerare le dipendenze di questi):

- *libc.so.**
- *ld-linux.so*
- *libext2fs.so.**
- *libcom_err.so.**

Copiamo le librerie ricorrendo al solito *'cp':*

```
cp -dp /lib/libreria_da_copiare /home/bkane/lib/
```

All'interno dell direcotry *'/etc'* inseriremo tutti i file di configurazione di cui avremmo bisogno (ovviamente più servizi decidiamo di utilizzare maggiori saranno i file di configurazione da creare). Mantenendo una struttura similare a quella di tutte le distribuzioni dobbiamo generare i file: *passwd, group* e *profile* (è possibile copiare direttamente il file della distribuzione fissa, adattandolo leggermente), *fstab* (file contenente informazioni indicative sui filesystem), *'termcap'* (file contenente le caratteristiche del terminale) e *'shells'* (file contenete tutte le shell utilizzabili). Generiamo ora il file *'ld.so.cache', cache* necessaria alle nostre librerie.

Ricordo che il file deve essere adattato alla nostra baby distribuzione, quindi utilizziamo il comando:

```
ldconfig -r /bkane/linux
```

Aggiungiamo ora la shell da utilizzare e i comandi che vogliamo siano disponibili. Come shell non utilizzeremo la *bash* (troppo voluminosa per i miei gusti) ma la

Figura 3

Librerie necessarie al funzionamento del sistema.

'ash' (operazione da fare dopo aver generato la directory *'bin'*), contestualmente generiamo un link a *'sh'* (il kernel prende a considerazione questo file per l'identificazione della shell):

```
cp /bin/ash /bkane/linux/bin
ln -s ash /bkane/linux/bin/sh
```

Ricordiamoci di modificare il file *'fstab'* copiato. Il file dovrà essere del tipo:

```
ramdisk / ext2 defaults 1 1
proc / proc proc defaults 0 0
```

Dove: il primo campo *(proc, ramdisk)* definisce il tipo di dispositivo o il file system da montare, il secondo definisce la directory che funge da punto di innesto per il file system, il terzo campo definisce il tipo di file system, il quarto campo descrive alcune opzioni speciali utilizzabili per il tipo di montaggio che si intende eseguire, il quinto identifica i file system che possono essere utilizzati per il dump, infine, il sesto campo viene utilizzato dal programma *'fsck'* per determinare l'ordine di controllo dell'integrità dei file system (leggere *'Appunti di Informatica Libera'* per avere maggiori informazioni a riguardo).
Copiamo ora i comandi che ci interessano.
Una soluzione potrebbe essere (ripetere l'operazione di *'cp'* per i file di nostro interesse):

- *cat:* comando utile a concatenare e stampare file sullo standard output;
- *chmod:* per cambiare i permessi a file/directory;
- *chown:* per cambiare: utente/gruppo proprietario di un file/directory;
- *cp:* copiare file;
- *gzip:* comprimere in formato *'.gz'*;
- *bzip2:* comprimere in formato '.bz2';
- *tar:* comprimere in formato '.tar';
- *loadkeys:* carica la tabella della tastiera (italiana, americana..);
- *ls:* stampa contenuto di una directory;
- *mkdir:* crea una nuova directory;
- *rm:* rimuove file/dirctory;
- *pico (in alternativa 'vi'):* editor di testo (necessario per modificare configurazioni o creare nuovi file);
- *sync:* necessario perché ci consente di scaricare nei dischi i dati contenuti nella memoria cache. Solitamente viene utilizzato durante la fase di *'halt'* del sistema per garantire che tutti i dati vengano registrati/terminati in maniera opportuna sul disco prima dello spegnimento del computer.
- *update: 'bdflush'*, necessario per inizializzare i demoni allo start del sistema.

Inseriamo all'interno dell directory *'/sbin/:*

- *mount:* necessario a montare i file system;
- *umount:* necessario per "smontare" i file system;
- *e2fsck:* check di un file system di tipo Linux *ext2;*
- *fdisk:* manipolare la tabella delle partizioni;
- *fsck:* controlla e ripara un file system Linux;
- *mkdosfs:* creazione di un file system dos;
- *mke2fs:* creazione di un file system Linux tipo *ext2;*
- *mknod:* creazione device a carattere o a blocchi;

All'interno dell directory radice generiamo un file *'linuxrc'* che verrà letto in fase di avvio del sistema. All'interno di questo file inseriremo:

```
#!/bin/sh
PATH=/sbin:/bin:/usr/
mount /proc/ /proc -v -t proc
loadkeys us.map
    (caricamento tastiera americana)
cd /
exec ash -rcfile /etc/profile
```

Non rimane che creare le directory di: *'/root/*, *'/proc'*, *'/mnt'*, *'/tmp'* e *'/usr'*.
"Smontiamo" il file system *'umount initrd'* e comprimiamolo:

```
gzip -9 initrd
```

Non rimane che copiare l'immagine su di un floppy, et volià, abbiamo messo alla luce un bel pargoletto! Ovviamente il supporto per hardware di ogni tipo necessita della ricompilazione del kernel! Qualora non aveste voglia di crearvi la vostra distribuzione è possibile utilizzare altre mini-distribuzioni presenti all'interno della Rete. Tra i progetti più seguiti spicca *LRP* (linux router project). Possibile scaricare la distribuzione (sempre su floppy) all'url: *www.lrp.org* La quantità di documentazione disponibile in linea sul progetto rende il tutto molto più appetibile ad utenti finali (possibile scaricarsi ed integrare, conforme alle proprie esigenze, una molteplicità di moduli e una molteplicità di immagini del kernel, sia della famiglia 2.2.x che della 2.4.x –con supporto pppoe-).
Dimenticavo, prima di spegenere la macchina ricordiamoci di digitare:

```
sync|umount -a -v -n|kill 1
```

Alla vostra fantasia la generazione di uno script più strutturato!

Francesco Tornieri

RIFERIMENTI

- Appunti di Informatica Libera.
- Documentazione del Kernel.
- Boot-disk Howto.
- Man on-line.

Il fatto che i server installati su Internet e che utilizzano Linux siano in costante aumento la dice lunga sulle potenzialità e sull'affidabilità di questo sistema operativo. Spesso però non si fa abbastanza attenzione all'ottimizzazione dell'hardware a disposizione che non viene sfruttato al meglio, o i demoni di rete lasciati con le impostazioni di default...

Massime prestazioni per i nostri server Web!!

Le principali distribuzioni Linux disponibili, includono una serie di servizi di rete che vanno dal web server Apache al gestore dns Bind, al demone smtp Sendmail, al server ftp

Kernel

Tutti questi servizi sono di fondamentale importanza in un server di rete che deve essere connesso ad Internet, e per questo motivo è sempre consigliabile ricompilare i sorgenti di questi pacchetti, così da avere in primis tutte le ultimissime versioni con eventuali bug di sicurezza corretti, e poi ricompilando, tutto il compilatore utilizzerà dei parametri specifici per l'hardware presente nella nostra postazione. Infatti, i pacchetti precompilati girano perfettamente sia su un i386, che su un i486 o un Pentium, ma se noi stiamo utilizzando un Pentium 4 probabilmente non sfrutteremo al top la nostra configurazione hardware. Quindi, il primo consiglio è proprio quello di ricompilare sempre i codici sorgenti, facendo attenzione a scaricare tali codici dai siti ufficiali. Eventualmente è opportuno controllare sugli appositi siti che si occupano di sicurezza informatica se per la versione che andremo ad installare esistono bug o exploit, ed eventualmente patch da installare. Ovviamente il primo passo obbligatorio è sempre quello di ricompilare il Kernel!

Apache

Il server web più utilizzato su Internet è sicuramente Apache. Apache, mette a disposizione il codice sorgente, così da poter entrare nei dettagli implementativi usati dai programmatori. Spulciando nella documentazione ufficiale sul sito *www.apache.org*, ci accorgiamo che fra gli obiettivi primari degli sviluppatori legati al progetto c'è innanzitutto la correttezza del codice e la sua robustezza, e subito dopo l'efficienza. Sulla sicurezza del codice non c'è nulla da ridire, infatti questo è il punto di forza di Apache. Se consideriamo poi che la maggior parte dei server Web è connessa ad Internet con accessi inferiori ai 10Mb/s, la letteratura presente in materia ci suggerisce che un comune PC con un processore Pentium o equivalente può gestirne tranquillamente le richieste. La cosa cambia se aumenta l'ampiezza di banda, e se i servizi da supportare comprendono elaborazioni di CGI e accesso a database. In questo caso diventa necessario disporre di hardware più performante ed eventualmente di un cluster per far fronte ad incidenti che blocchino il servizio.

Considerazioni Hardware

Il componente hardware di maggior interesse e peso sulle prestazioni di un server web è senza dubbio la RAM. Infatti un server web non dovrebbe mai swappare sugli harddisk per rispondere alle richieste. Questo comporta un'attesa

maggiore da parte dei client che devono "attendere" il ricarico dei dati dalla zona di swap. Per evitare questo o si aumenta a sufficienza la Ram del sistema, o si imposta a valore più basso il parametro Max Client, così da evitare di sommergere di richieste il server. Subito dopo le considerazioni sulla memoria del server i componenti da ottimizzare sono gli adattori di rete e gli hard disk. Su questi componenti la pratica e l'esperienza vi guideranno nella scelta. Sono infatti disponibili diversi fornitori di schede di rete, di hard disk, di hub, switch, etc, ma in base alla situazione specifica la scelta potrebbe essere diversa.

Configurazioni Run-Time

Il file di configurazione principale del web server Apache è *httpd.conf*, e si trova di solito nella directory *conf* del path scelto con l'opzione *—prefix* durante la compilazione. Tale file appare come un semplice file di testo, su ogni riga va specificata un'opzione, se la linea inizia con un carattere # allora viene ignorata, perché considerata commento. Ci possono essere commenti lunghi a dismisura, questo non influenzerà le prestazioni, anzi è consigliabile utilizzare note il più possibile esplicative sulle modifiche apportate a tale file per permettere un'agevole manutenzione futura. La prima direttiva che analizziamo è:

```
HostnameLookups
```

Tale direttiva implica che ad ogni richiesta ricevuta dal server corrisponda una richiesta al dns per cercare di risolvere l'hostname legato all'IP del client. E' ovvio che questa procedura non può che costringere ad un'attesa affinché il dns risolva la query e restituisca la risposta. Nelle versioni precedenti alla 1.3, l'opzione era abilitata di default, ma nelle ultime release è già impostata a Off. Bisogna però fare attenzione a non specificare nessuna direttiva Allow from o Deny from con un nome di dominio, o altrimenti bisognerà aspettare affinché due query al dns abbiano luogo. Si può comunque essere selettivi su quali richieste e quali tipi di file debbano essere trattati con la direttiva *HostnameLookups* impostata a *On*.

```
HostnameLookups off
<Files ~ "\.(html|cgi)$">
    HostnameLookups on
</Files>
```

Per esempio il codice precedente abilita la risoluzione dei nomi solo per i file con estensione *.html* o *.cgi*. Nel caso in cui sia necessario, invece, avere a disposizione il nome degli host collegati al server web, ad esempio per generare delle statistiche o dei report geografici, è sempre possibile utilizzare dei programmi ad hoc che in un secondo tempo risolvano gli IP. Un esempio pratico è l'utilizzo di logresolve, soluzione sviluppata ad hoc per questo scopo.
In effetti è presente in questo software una tabella hash in cache per limitare l'impatto che avrebbe la sua elaborazione sul DNS della rete.

Figura 1

L'home page del sito www.apache.org.

AllowOverride

Questa direttiva permette un controllo sull'accesso a particolari file e zone del server. L'unico problema è che la ricerca del file di configurazione, di solito .htaccess è ripetuta più volte, una per ogni parte del path della DocumentRoot.
Esempio:

```
DocumentRoot
                /www/htdocs
<Directory />
    AllowOverride all
</Directory>
```

Se l'URI richiesto fosse */index.html* allora Apache inizierebbe prima con il cercare */.htaccess*, poi */www/.htaccess*, poi di nuovo */www/htdocs/.htaccess*. Questo fa capire che è molto più efficiente settare l'intero filesystem con l'opzione:

```
AllowOverride None
```

E solo le directory e i file effettivamente necessari con sottosezioni con la direttiva

```
AllowOverride None.
```

DirectoryIndex

Questa direttiva informa il web server su quale file servire al client in caso di richiesta incompleta sul nome specifico del file. Ad esempio se noi scriviamo nel browser *www.sitovisitato.com,* noi non stiamo richiedendo alcuna pagina in particolare, perché abbiamo solo indicato il nome del server. Tuttavia nel 99% dei casi ci compare l'Home Page del sito. È appunto la direttiva che stiamo trattando che si occupa di fornire al client la pagina principale in caso di richieste incomplete. Tuttavia questa opzione permette anche si specificare delle wildcard, ad esempio:

```
DirectoryIndex index
```

Al posto di:

```
DirectoryIndex index.
    cgi index.pl index.shtml
                    index.html
```

Nel primo caso noi non indichiamo quale versione della pagina index servire, mentre nel secondo caso stabiliamo una gerarchia precisa su quale pagina mandare al client.

Creazione di processi

Prima della versione 1.3 di Apache le direttive *MinSpare-*

Servers, MaxSpareServers e *StartServers* avevano un forte impatto sia sulle prestazioni che sui risultati generati dai benchmark. Questo perché una scelta implementativi limitava la creazione di un nuovo processo figlio ogni secondo. Questo era stato fatto per evitare di inondare di richieste contemporanee il server.
Ad esempio se i server iniziali sono impostati a 5 (StartServers 5), e all'improvviso arrivano 100 richieste, allora sono necessari 95 secondi affinché siano creati tutti i processi figli. Questo limite è stato superato nelle nuove versioni, perché la regola di un nuovo processo al secondo è stata cambiata con una regola esponenziale, cioè viene generato un processo figlio il primo secondo, 2 il secondo secondo, 4 il terzo secondo, 8 il quarto e così fino a 32 processi. La progressione si interrompe non appena si raggiunge il valore impostato come *MinSpareServers*.
Le altre direttive collegate a questo ambito sono *MaxRequestPerChild* e KeepAliveTimeout. Nello specifico la prima indica quante richieste al massimo possono essere gestite da un singolo processo figlio. Il valore di default è 0 che vuol dire infinite. Se invece settassimo questo valore basso potremmo peggiorare le prestazioni del server. La seconda opzione indica invece per quanti secondi deve rimanere in attesa il processo figlio di una nuova richiesta sulla stessa connessione, questo per evitare di ricrearlo nuovamente.
Tale valore deve essere impostato in base all'ampiezza di banda e al traffico del server. Il valore predefinito è di 15 secondi. Il fatto da sottolineare è che per questo tempo il processo figlio in questione non potrà essere utilizzato per gestire richieste diverse provenienti da altri client.

Opzioni di compilazione mod_status

Compilando questo modulo all'interno di Apache e settando la direttiva

```
ExtendedStatus On
```

Noi forziamo l'esecuzione di due system call ad ogni richiesta ricevuta dal server. Infatti noi abbiamo scelto di avere attivata la modalità di debug che include la chiamata alla funzione di sistema che ci ritorna la data e l'ora, che sono poi incluse nel file di log generato. Di default tale direttiva è disabilitata e dovrebbe essere usata solo in fase di test del sistema.

```
ScoreBoard File
```

Il processo padre e i processi figli di Apache comunicano fra di loro attraverso un meccanismo chiamato ScoreBoard. Tale meccanismo dovrebbe essere implementato utilizzando la memoria condivisa (Shared Memory). In effetti è questa la strada utilizzata per i sistemi operativi che supportano questo meccanismo.
La scelta implementativa di default per gli altri sistemi operativi è basata sull'utilizzo di un file all'interno dell'Hard Disk. E' facilmente comprensibile come questo sia più lento e meno affidabile.
La soluzione per risolvere questo problema o per controllare la strategia usata è controllare il file

```
src/main/conf.h
```

che si trova nella directory dove abbiamo compattato i sorgenti relativamente alla nostra architettura hardware. In tale file dobbiamo cercare

```
USE_MMAP_SCOREBOARD
USE_SHMGET_
        SCOREBOARD
```

Definendo una di queste due opzioni assieme a *HAVE_MMAP* e *HAVE_SHMGET*, abilitiamo l'utilizzo del codice con la memoria condivisa.

Dynamic_ Module_Limit

Se non è nostra intenzione utilizzare i moduli dinamici cariati all'interno di Apache è possibile specificare 0 come valore per tale variabile a compile time, così da risparmiare memoria ram che sarebbe altresì utilizzata per supportare il caricamento dinamico dei moduli.

Considerazioni sul modello Pre-Forking

Il modello utilizzato sulle piattaforme Unix-Like di Apache è di tipo Pre-Forking.
Questo vuol dire che il processo padre non si occupa mai di servire le richieste che arrivano dai vari client, ma il suo unico compito è quello di allocare processi figli in numero necessario a soddisfare le richieste, o a mandare loro il segnale di KILL se necessario.
E' infatti compito dei processi figli elaborare la risposta da mandare ai client.
Questo modello offre un grado di robustezza unico sull'affidabilità del servizio.
Così facendo il codice del processo padre rimane semplice e snello, mentre quello dei figli è più elaborato, specialmente nel caso di elaborazione di CGI o altro.
Però così facendo, anche in caso di errore nell'elaborazione di una risposta, il processo padre continuerà a lavorare sempre, mentre uno o più processi figli potrebbero bloccarsi o andare in segmentation fault.
In tal caso il processo padre, monitorando lo ScoreBoard manderà il segnale di KILL e provvederà a riallocare altri processi nuovi.
Questa strategia è stata uno dei principali punti di forza di Apache, e certamente non verrà cambiata in futuro.
Comunque non mancano certo le critiche a questo modello che viene accusato di causare un eccessivo context switch fra i processi o un sovraccarico nell'utilizzo della memoria che deve mantenere allocati numerosi processi. Inoltre non è possibile implementare un meccanismo efficiente per il caching delle richieste.

Conclusioni

La pratica e l'esperienza nell'utilizzo dei sistemi informatici complessi unitamente ad una buona documentazione di base sono comunque sempre la migliore strada da seguire per ottenere le massime prestazioni dai nostri server.
L'aspetto dell'ottimizzazione delle risorse non è mai da trascurare, perché può portare ad un considerevole risparmio in termini economici, o comunque ad un aumento dello standard qualitativo del servizio.

Carlo Stumpo

ADSL e' l'acronimo di Asymmetric Digital Subscriber Loop, un nuovo tipo di accesso digitale, oramai, punto di forza di tutti i gestori telefonici. I vantaggi della nuova tecnologia?

- Accesso alla rete attraverso il normale doppino telefonico (standard nell'impianto telefonico presente nelle abitazioni)
- Trasmissione dati ad alta frequenza (superiore ai 32kHz), consentendo la gestione contemporanea sia di un servizio di fonia che di un servizio di trasmissione dati
- Nessun costo telefonico aggiuntivo dato dalla tariffa telefonica oraria (pagamento di un canone fisso)
- Connessione 24 ore su 24

L'Adsl e' stata pensata per applicazioni SOHO (Small Office Home), per essere utilizzata all'interno di piccole reti (uffici/case).

ADSL: Linux e l'alta velocità

Stufi dei lunghi tempi di download? Soluzione? Velocizziamo la connessione...poi? Come permettere al nostro sistema Linux di interagire con la nuova tecnologia? Analizziamolo insieme...

Per cominciare...

Dopo essersi accertati che la nostra abitazione disponga di una rete telefonica analogica (se utilizzate linea isdn, il gestore telefonico dovrà necessariamente installarvi una linea analogica per la gestione del nuovo servizio), dobbiamo considerare il tipo di hardware che si pensa di utilizzare. Qualora volessimo comperare un modem ADSL e' opportuno verificare l'effettiva compatibilità (esistenza di driver della casa produttrice) di questo con Linux. Solitamente la connessione tra PC e modem ADSL avviene in due modi:

- tramite porta usb;
- tramite scheda ethernet.

L'utilizzo di una periferica usb impone di porre particolare attenzione alla versione di kernel utilizzata dal nostro sistema:

- in distribuzioni utilizzanti kernel della famiglia 2.2.X e' opportuno utilizzare alcune patch disponibili all'url ' *www.linux-usb.org*';
- in distribuzioni utilizzanti kernel 2.4.x non avremmo problemi visto che il codice per la gestione di periferiche usb e' stato riscritto e migliorato.

Nessun problema se la connessione passa attraverso l'utilizzo di una scheda di rete.
Ultimamente e' uso di molti gestori telefonici dare in comodato d'uso modem ADSL (solitamente della famiglia Alcatel Speed Touch). Ad una prima vista il modem presenta:

- alimentazione esterna;
- scheda ethernet 10baseT (RJ45).

Attualmente i gestori telefonici propongono abbonamenti che possono avere una velocità variabile di 640 kbit/s o 256 kbit/s in entrata e di 128 k/bit in uscita (5-10

Figura 1
Modem della famiglia Alcatel Speed Touch -modello Home-.

volte più veloce di una normale connessione analogica a 56k). In questo articolo prendiamo in considerazione il caso in cui il gestore telefonico ci offrirà, in comodato d'uso, il modem (offerta Telecom ADSL light, utilizzo di modem Alcatel Speed Touch Home).
Il comodato d'uso e' un contratto con il quale una parte (gestore telefonico) consegna all'altra (utente sottoscrittore contratto) una cosa mobile/immobile (modem ADSL) affinché se ne serva per un arco temporale determinato con l'obbligo di restituire la stessa a termine (scadenza contratto). La soluzione considerata prevede la connessione tra personal computer e modem attraverso una scheda di rete.

Installazione

Preliminarmente dovremmo assicurarci che il sistema Linux riconosca la scheda ethernet in nostro possesso. Per accertarci di questo, da console, digitiamo:

```
blackkane# ifconfig

eth0 Link encap:
        Ethernet HWaddr
        00:60:08:0F:A8:EC
 inet addr:192.168.1.2
        Bcast:192.168.1.255
        Mask:255.255.255.0
 UP BROADCAST
        RUNNING MULTICAST
        MTU:1500 Metric:1
 RX packets:17551
 errors:0 dropped:0 over-
        runs:0 frame:0
 TX packets:13533
 errors:0 dropped:0 over-
        runs:0 carrier:6
 collisions:1
        txqueuelen:100
 RX bytes:20753686
        (19.7 Mb) TX
 bytes:1218569 (1.1 Mb)
 Interrupt:7 Base
        address:0xb000
```

Qualora l'output risultante non fosse del tipo riportato sopra, dovremmo ricompilare il kernel.
Portiamoci all'interno della directory di sistema contenente i sorgenti del kernel *'/usr/src/linux'*, e digitiamo:

```
make xconfig (modalita'
        grafica)
make menuconfig
        (modalita' testo)
```

Fatto questo abiliteremo:

- *Network device Support:* necessario abilitare la voce se si vuole utilizzare una scheda ethernet e il protocollo di comunicazione ppp.
- *ppp:* 'Point to Point' utile per inviare il traffico Internet attraverso la linea telefonica. Possibile utilizzare come modulo il driver *(ppp_generic.o).*
- *Ethernet 10-100:* necessario evidenziare la voce per consentire al sistema di utilizzare una scheda ethernet (e' stata utilizzata una *3c509 etherlink III* della *3com*).

Infine digitiamo:

```
make dep; make clean;
        make bzImage
```

seguito da:

```
make modules;
        make modules_install
```

L'iter seguito porterà alla creazione di un nuova immagine contenente le feature necessarie. Dopo aver copiato questa all'interno della directory *'/boot'* e' necessario editare il file *'/etc/lilo.conf'*, ed aggiungere:

```
image  = /boot/slack
label  = slack
root   = /dev/hda7
initrd = /boot/initrd
```

Dove, con il termine:

- *image,* definiamo la path corretta dell'immagine creata *(/boot/slack)*;
- *label,* definiamo il nome con il quale potremmo, ad avvli successivi, richiamare la nuova immagine;
- *root,* definiamo la partizione utilizzata come mount point;
- *initrd,* definiamo l'initial ramdisk (il kernel linux offre la possibilità di caricare un minuto filesystem in una ramdisk ed eseguire da qui dei programmi prima che il filesystem venga montato).

Salviamo e rilanciamo il lilo (in questo modo al prossimo riavvio verrà riconosciuta l'immagine appena creata):

```
lilo -v
Reading boot sector from
/dev/hda
Merging with /boot/boot.b
Mapping message file
/boot/message
Boot image: /boot/slack
Mapping RAM disk
/boot/initrd
Added slack *
```

Sfortunatamente l'utilizzo del demone *'ppp'*, solitamente il solo necessario in connessione analogiche, non sarà sufficiente. Per gestire il nuovo tipo di connessione dobbiamo ricorrere al protocollo *PPPoE (Point-to-Point Protocol over Ethernet). PPPoE* e' un protocollo che consente di incapsulare frames PPP in ethernet frames. E' possibile scaricare il pacchetto dall'url *'www.roraring-penguin. com/pppoe'*. Il protocollo PPPoE necessita di:

- sistema linux avente kernel della serie: 2.0-2.2-2.4;
- supporto per *'packet socket'* (abilitazione nel kernel):il protocollo Packet e' utilizzato da applicazioni che comunicano direttamente con device di rete non aventi protocollo implementato nel kernel. Possibile utilizzare in maniera modulare i 'packet soket' ricorrendo al driver *'af_packet.o';*
- demone pppd 2.3.7 (o superiori);

Dopo aver scaricato il pacchetto PPPoE nel formato di compressione più consono alle nostre esigenze (e' disponibili sia in formato *'tgz'* che *'rpm'*), non rimane che installarlo. Qualora avessimo scaricato la versione avente suffisso *'.rpm'* digiteremo (il tutto e' stato testato utilizzando una distribuzione

Suse 7.1):

```
rpm -Uhv rp-pppoe-3.2.1-1
      .i386.rpm /usr/sbin/
        adsl-setup (nessuna
              gui grafica)
```

```
rpm -Uhv rp-pppoe-3.2.1-1
      .i386.rpm rp-pppoe-gui-
          3.2.1-1.i386.rpm
          /usr/bin/tkpppoe
           (con gui grafica)
```

Alternativamente (suffisso file *'.tar.gz'*) digiteremo:

```
tar zxvf rp-pppoe-3.2
                   .tar.gz
```

Verrà creata la directory *'rp-pppoe-3.2'* contenente due file eseguibili:

- *go:* se non si vuole nessuna GUI;
- *go-gui:* se si volesse interfaccia grafica (testato con Slackware 8.0 e Suse 7.1);

All'interno del pacchetto sono inclusi:

- *sniffer:* in modo da esaminare le connessioni in essere e decidere il 'modus operandi' di pppoe;
- *server:* modalità server (maggiori informazioni all'interno del sito);
- *relay agent:* per gestire sessioni di *pppoe* multiple da piu' macchine dietro un relay utilizzando opportune patch *(http://www.shoshin.uwaterloo.ca/~mostrows/)*, abilitare il supporto di *'pppoe'* nel kernel (kernel della famiglia 2.4.x).

Dopo aver ricompilato il kernel (controllare di aver abilitato sia la voce *'PPP over Ethernet'* sia la voce *'Packet Socket'*), e' necessario includere, all'interno del file di configurazione di *ppp* *(/etc/ppp/options)*, la stringa:

```
plugin /usr/lib/pppd/
  plugins/pppoe.so (control-
lare l'url 'http://www.sho-
shin.uwaterloo.ca/~mostro
ws/' qualora si riscontras-
        se qualche problema)
```

Infine inizializzare il demone *'pppd'* digitando:

```
pppd eth0 (accertarsi che
     l'interfaccia di rete sia
         attiva utilizzando il
              comando ifconfig)
```

TKPPPoE

L'utilizzo della GUI consentirà una più facile gestione della connessione *adsl*. Il fornt-end e' basato sulle librerie tcl/tk (rev. 8.0), accertarsi, quindi, che le suddette librerie siano installate all'interno del sistema. Digitare da shell:

```
less /var/log/packages/tcl
| grep version
```

Ottenendo un output simile a:

```
tcl: The Tcl script
   language, version 8.3.3.
```

L'utilizzo della GUI ci permetterà di:

- definire il tipo di connessione (solo l'utente root può fare questo);
- iniziare/terminare una sessione di connessione.

Per lanciare l'interfaccia grafica digitare:

```
tkpppoe
```

Apparirà in finestra (indispensabile avere preliminarmente aperto una sessione X) un front-end come quello riportato in Figura 2. Non avendo definito nessun tipo di connessione e' necessario cliccare sul pulsante *'New Connection'*. Ci troveremo a dover inserire informazioni sulla nostra connessione. La nuova finestra (Figura 3) sarà divisa in quattro sezioni distinte:

1 - Basic
2 - Nic and Dns
3 - Options
4 - Advanced

Figura 2

Front-end grafico tkpppoe.

All'interno della sezione 'Basic' dovremmo inserire (Figura 4):

- *Connection name:* campo identificativo del nome della connessione (esempio:blackkane).
- *User Name:* username/login assegnatoci dal gestore telefonico/ISP.
- *Network:* dominio del gestore telefonico (alcuni provider aggiungono questo campo alla username).
- *Password:* password assegnataci dal gestore telefonico/provider.

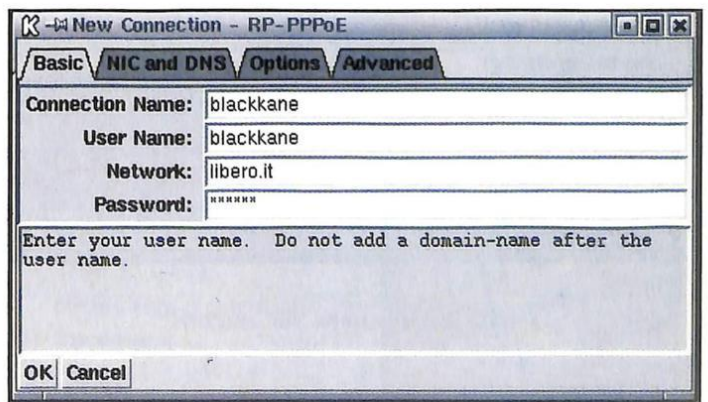



Figura 3

Campi da completare per la sezione 'Basic'.

All'interno della sezione *'NIC and DNS'* dovremmo (Figura

5) riempire sia il campo *'ethernet interface'* (interfaccia di rete connessa al modem *adsl (-eth0, eth1, ethN-)* sia quello *'DNS setup'*.
La sotto sessione *'DNS setup'* mette a disposizione tre differenti opzioni:

1 - *'From Server'*: caso in cui il sistema otterrà le informazioni di DNS direttamente dal server *PPPoE;*

2 - *'Specify'*: caso in cui si disponga di una server DNS interno alla propria lan, inserire manualmente indirizzo IP (inserire indirizzo ip di DNS primario e secondario);

3-' *Do not Adjust'*: utilizzare qualora non fosse necessario utilizzare server DNS.

Figura 4

Campi da completare per la sezione 'NIC and DNS'.

Nella sezione *'Options'* dovremmo completare i campi (Figura 6):

- *Allow use by non-root users:* abilitare l'opzione se si vuole che normali utenti possano inizializzare/ terminare la connessione internet.

- *Use synchronous PPP:* da abilitare per evitare eccessivi carichi di lavoro della CPU.

La sotto sessione *'Firewalling'* permette di configurare il nostro server ADSL in tre differenti modi:

1- *Stan-Alone:* installazione di semplici regole di firewalling (da usare se il computer avente il modem ADSL non fa parte di nessuna rete).

2- *Masquarading:* semplici regole di firewalling che permettono alla nostra macchina di fungere da gateway per la nostra lan.

3- *None:* abilitare questa opzione se sono gia' attive all'interno delle nostra rete delle regole di firewalling (nel seguito vedremo come configurare il firewall utilizzando il pacchetto *'Susefirewall'*).

La sessione *'Advanced'* puo' non essere considerata (Figura 7), visto che solo alcune connessioni ADSL (nessuna italiana) necessitano delle informazioni contenute in questa sessione (*'Service-Name'* e *'AC-Name'*).

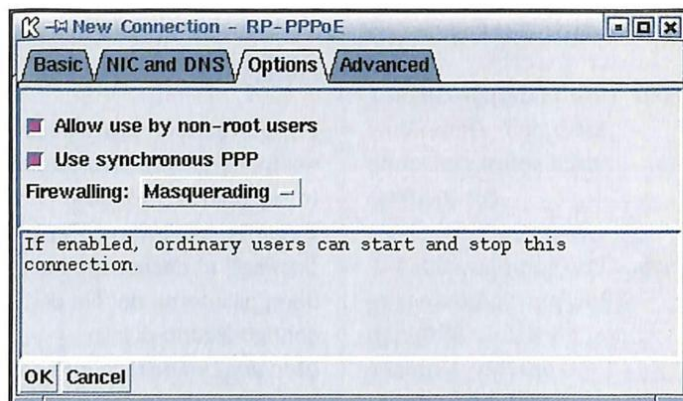



Figura 5

Per creare una lan domestica (nessun firewall attivo) è indispensabile abilitare la voce 'Masquarading'.

Configurata la connessione non rimane che digitare:

```
tkpppoe
```

Figura 6

Solitamente non vengono completati i campi della sezione 'Advanced' (contattare il gestore telefonico per sapere se questi campi devono essere riempiti).

La GUI che ci apparirà sarà del tipo riportato in Figura 8: non rimane che cliccare il pulsante *Start*. E' possibile controllare lo stato della connessione visionando i led presenti nella GUI (controllare la documentazione per avere una rapida guida ai differenti colori dei led).

Suse-Firewall

La connessione e' configurata. Paghiamo un canone fisso. Possiamo tranquillamente lasciare il nostro PC accesso e connesso 24 ore al giorno. Sicuri di non volere proteggere in alcun modo il nostro sistema con qualche policy di sicurezza? Nessuna pratica con *ipchains/iptables*? No? Poco male. Suse viene incontro ai suoi utilizzatori

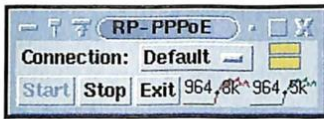



Figura 7

Dopo aver inizializzato la connessione verrà mostrata un front-end simile a quello riportato nella foto.

con un pacchetto contenente un filtro: *Susefirewall.* Cosa si intende per 'filtro di pacchetto'? Un *packetfilter-firewall* permette all'amministratore di proteggere la propria rete contro accessi non espressamente attivati. L'utilizzo di questa utility della Suse e' da usarsi solo nel caso di reti domestiche o di reti non aventi bisogno di utilizzare policy particolarmente restrittive di sicurezza (il filtro di pacchetto non e' mai sostitutivo di un firewall). Presupponendo di aver selezionato la voce *'None'* nella sezione *'Firewalling'* della GUI *tkpppoe* non rimane che iniziare a dare un'occhiata al file *'/etc/rc.firewall'* (importante ricordare di attivare il masquerading qualora la nostra macchina server dovesse permettere ad altri PC di connettersi ad internet).
All'intero di *'rc.firewall'* troveremo le direttive (verranno considerate solo le direttive di uso più comune):

- *FW_START:* da abilitare *('yes')* per inizializzare lo script (necessario per lanciare firewall e masquarading). Possibile inizializzare il filtro di pacchetto ad ogni avvio, abilitando la voce corrispondente all'interno del file *'rc.config';*

- *FW_DEV_WORLD:* inserire il dispositivo che ci consente di uscire su Interne (nel caso di connessione ADSL, dovremmo inserire l'interfaccia ethernet che ci permette questo - *ex:eth0-*);

- *FW_DEV_INT:* interfaccia di rete adibita allo scambio di informazioni all'interno della lan *(ex:eth1);*

- *FW_ROUTE:* abilitare la voce per permettere al sistema di 'mascherare' altri PC all'interno della rete. Ricordo che abilitando quasta opzione non potranno essere contattati dall'esterno PC interni alla lan, avendo questi indirizzi ip appartenenti a reti private *(ex:192.168.1.2)* non venendo, qundi, instradati (routed);

- *FW_MASQUARADE:* attivare masqurading;

- *FW_MASQ_NETS:* inserire l'ip indicativo della rete sulla quale deve esse attivato il masquarading *(ex:192.168.1.0/8);*

- *FW_PROTEC_FROM_INTERNAL:* attivare questa voce se si vuole proteggere la macchina che funge da firewall da possibili attacchi provenienti dalla rete interna (inutile abilitare la voce se siamo gli unici utenti della nostra rete);

- *FW_SERVICES_EXTERNAL_TCP/FW_SERVICES_EXTERNAL_UDP:* inserire all'interno di questa voce i servizi (esterni) tcp/udp a cui si deve poter accedere *(ex:21 22 23);*

- *FW_SERVICES_INTERNAL_TCP/FW_SERVICES_INTERNAL_UDP:* inserire i servizi tcp/udp a cui si deve poter accedere dalla rete interna *(ex:22 80);*

- FW_ALLOW_INCOMING_HIGHPORTS_TCP/FW_ALLOW_INCOMING_HIGHPORTS_UDP: permesso di accesso a connessioni in entrata su porte elevate;

- *FW_LOG:* settare livello di log del sistema;

- *FW_KERNEL_SECURITY:* abilitazione feature del kernel addizionali per la sicurezza del protocollo TCP/IP;

- *FW_ALLOW_PING_FW:* permettere o meno *ping* remoti;

- *FW_ALLOW_FW_TRACEROUTE:* permettere o meno *traceroute;*

- *FW_MASQ_MODULES:* utilizzo aggiuntivo di moduli per il *masquarading* (necessario abilitare la voce qualora si volesse utilizzare un server linux per la gestione di PC client pee il gioco on-line -quake-).

Ulteriori policy (avanzate) possono essere impostate abilitando la voce:

```
'FW_CUSTOMRULES'
        (file '/etc/rc.config.d/
              firewall-custom.rc.
                      config/).
```

Qualora si volesse abilitare questa voce e' consigliato aggiornare il file scaricando l'upgrade dello stesso dall'ftp Suse:

```
rpm -U ftp://ftp.suse.com/
            pub/suse/i386/
          update/7.1/secl/
      personal-firewall.rpm
```

In questo modo verrà fatto un aggiornamento in automatico del file (upgradare anche il pacchetto *'Susefirewall'* nello stesso modo. Possibile visionare una serie di esempi dell'uso del pacchetto *'Susefirewall'* leggendo il file *'usr/share/doc/packages/SuSEfirewall/EXAMPLE'*.
Che altro dire?
Ci si vede in Rete!!

Francesco Tornieri

Riferimenti

- *A PPPoE Implementatio for Linux* di David F.Skoll.
- *PPPoE and Linux (slide)* di David F.Skoll.
- *ADSL How-to per sistemi Linux* di David Fannin.
- *PPPoE (www.roaringpenguin.com/ pppoe).*
- *Tkpppoe - A GUI for managing PPPoE Connections (www.raringpenguin.com/ pppoe/tkpppoe).*
- *Manuale Suse Linux.*

Analisi comparativa dei Journal File System per GNU/Linux

"Cos'ha questa macchina? Si può usare?"
"Sa gli studenti... sono stati abituati a sedersi e schiacciare il bottone... quindi la macchina sta facendo il check del disco!"
"Sì, ma noi qui a scuola dobbiamo essere subito operativi... come con i286!..."

Il FileSystem è una componente chiave per assicurare prestazioni ed affidabilità in un ambiente per applicazioni critiche e perché no anche didattiche: siano esse per server che per client.
I requisiti per un filesystem di moderna concezione sono: capacità di supportare grossi volumi fisici (oramai centinaia di gigabyte), integrità dei dati, riduzione della frammentazione interna ed esterna, recovery veloce in caso di crash e manipolazione in modo veloce ed efficiente sia dei file voluminosi che piccoli.
Il vecchio filesystem UFS come lo ext2 non garantisce un adeguato supporto alle nuove tecnologie, e nelle esigenze di scalabilità in un server generalmente si avverte un calo di performance. Aziende come IBM e SGI hanno sviluppato per GNU/Linux i loro filesystem e le nuove distribuzioni già li supportano durante il processo di installazione.

Perché Journal?

La sostanziale differenza tra un filesystem journal (come lo JFS) ed uno non journal (come lo ext2) sta nella capacità di recuperare lo stato consistente dell'hard disk in pochi secondi. La tecnica di recovery, ereditata dai database, si basa sulle transazioni. In caso di "crash" del sistema, lo stato consistente del disco viene recuperato leggendo i log e riapplicando le modifiche: quindi lo stato precedente.
Contrariamente in un filesystem non journal lo stato consistente del disco passa attraverso l'analisi dei metadata (i-node, directory, i-node map...) che richiede la scansione completa (blocco per blocco) del volume logico.
Questo avviene in quanto la ricerca dei blocchi liberi in un filesystem tipo ext2 avviene mappandoli in una sequenza di bit (detta comunemente "struttura a bitmap"); più il filesystem cresce in dimensioni, più lo spazio occupato cresce conseguentemente: l'algoritmo di ricerca utilizzato è sequenziale. Sia chiaro che per filesystem di piccole dimensioni il sistema è decisamente efficiente, e non penalizza le prestazioni del server.
Gli i-node rappresentano oltre che i file e le directory anche qualsiasi altro oggetto del filesystem, e sono fonda-

mentali per la ricerca degli stessi. Nei filesystem "tradizionali" (come lo ext2) devono necessariamente essere fissati a priori in base alla quantità di oggetti che si suppone saranno presenti. Una volta raggiunto il limite, non sarà più possibile creare oggetti anche se lo spazio sul disco è fisicamente presente (intendo i gigabyte disponibili e non ancora utilizzati). Nei filesystem journal questo non accade: la tecnica di allocazione dinamica degli i-node (dynamic i-node allocation) permette tuttavia di superare il limite con disinvoltura, anche a scapito di una complessità interna nella gestione.
Un extent è una sequenza di blocchi contigui allocati da un file come unità base, al loro interno contengono una sequenza composta da: physical, logical offset, length. In buona sostanza: l'ultimo termine indica l'inizio del blocco, il secondo la sua lunghezza ed il terzo l'offset rispetto al primo byte. Esempio: un file può contenere come extent le terne:

<30, 60, 1> e *<500, 20, 61>* (il secondo extent incomincia laddove termina il primo).

Gli extent sono inseriti negli i-node per mappare gli oggetti del filesystem. L'uso di extent riduce il tempo di ricerca dei blocchi e la frammentazione del disco fisso. Congiunti alla tecnica B+Tree, ereditata anch'essa dai database, incrementa notevolmente le performance sia per la ricerca dei blocchi liberi, sia per la ricerca dei file nei directory. La gestione dei file sparse e dei file large o dense, cambia invece per ogni filesystem in esame.

Filesystem ext3

ext3 è il filesystem ufficialmente supportato da Red Hat Linux per la nuova versione 7.2. Non si tratta di una vera e propria evoluzione dello ext2, bensì di un *"add-on"* per dotarlo di caratteristiche journal: ossia accompagna i pregi e non migliora i difetti. Non è indispensabile riformattare la partizione, è sufficiente convertirla e la compatibilità è garantita: con i nuovi *efstools* è altresì possibile montare una partizione ext3 come una comunissima ext2. L'operazione per rendere journal un "antidiluviano" ext2 è molto semplice:

```
[root@localhost /root]#
        tune2fs -j /dev/hdaX
```

La partizione non deve essere smontata. Verrà creato un file *'.journal'* nella radice root. La seconda operazione è di disabilitare il check del disco ad intervalli regolari (ogni centottanta giorni o ad ogni venti riavvi o mount):

```
[root@localhost /root]#
        tune2fs -c 0 -i 0
                /dev/hdaX
```

Nel file *'/etc/fstab'* si dovrà porre al posto di *ext2* lo *ext3* oppure auto se si usa *'e2fsprogs'* alla versione almeno 1.23. La peculiarità dello *ext3* risiede nel poter scegliere tre modi diversi di journaling: *"ordered"*, *"journal"* e *"writeback"*. Il modo *"journal"* viene effettuato sia sui dati che su i metadati. Negli altri due casi il journaling è limitato solamente ai metadati. Il metodo più rischioso è lo *"writeback"* dove il sistema potrebbe apparire in uno stato consistente, mentre potrebbero essere presenti dati vecchi. Nel metodo *"ordered"* i dati vengono scritti prima dei metadati e garantisce che il filesystem sia in uno stato consistente. Per il journal sfrutta uno strato layer generico *'jbd'* che gli consente di lavorare con un il journal esterno al filesystem, come un device creato in RAM. Rimangono invariate le dimensione massime del filesystem a 4Tb e la dimensione massima di un file a 2Gb.

Figura 1

Al lavoro nel laboratorio Linux del "Besta"

Rimane la struttura a bitmap di ricerca dei blocchi liberi: efficiente per filesystem di piccole dimensioni, ma del tutto inadeguata per i filesystem di grosse dimensioni. Per questi ultimi è più efficiente una struttura ad albero, *B+Tree, HTree, ...* (usata anche per organizzare le directory) con l'uso congiunto degli extent. Questi ultimi consentono di raggruppare i blocchi liberi in modo più efficiente della struttura a bitmap (che assegna ad ogni blocco del disco un bit). L'algoritmo di scansione modificato ad albero è più efficiente di quello sequenziale.

Filesystem JFS

Il filesystem JFS (Journal File System) sviluppato da IBM per i suoi server AIX è stato ora rilasciato con licenza GPL e messo a disposizione per la comunità GNU/Linux. Subito si distingue per la gestione delle directory. Le directory vengono organizzate in due modi differenti in base alle dimensioni stesse: nel primo caso ne contengono fino ad otto ("." e ".." escluse), e la struttura viene immagazzinata nello i-node stesso, nel secondo caso, con più di otto directory, il tutto viene affidato all'algoritmo B+Tree. La struttura interna è a 64 bit come lo XFS di Silicon Graphics. I limiti del filesystem sono: minimo 16 Mbyte, 512 Terabyte (con blocchi da 512K) e 4 petabytes con blocchi di 4 Kb. Ricordiamo che il kernel 2.4 presenta questi limiti: 16 Terabyte (con blocchi di 4Kb) per i file e 2 Terabyte per il filesystem. La massima dimensione dei file è legata al virtual file system e comunque ha gli stessi limiti del kernel. Le utility per creare e gestire le partizioni si trovano nel pacchetto *'jfsprogs'*. Per creare un filesystem è sufficiente digitare:

```
[root@localhost /root]#
        mkfs.jfs /dev/hdbX
[root@localhost /root]#
     mount -t jfs /dev/hdbX
             /mnt/jfs_root
```

Non è possibile convertire la partizione *ext2* in una JFS, peraltro operazione possibile solo passando da *ext2* a *ext3*. Sugeriamo di:

- Ridimensionare una partizione precedente, oppure utilizzare un altro disco.
- Creare la partizione con il nuovo filesystem.
- Montarla in un directory del root filesystem.

```
[root@localhost/root]# find/
-mount -print | cpio
-dpm /mnt/root_jfs
```

Farlo per tutte le vecchie partizioni del vostro disco: tipo, se avete separata una partizione di home:

```
[root@localhost /root]# find/
home -mount -print
| cpio -dpm
/mnt/root_jfs
```

e così via...
L'operazione di copia richiede molto tempo, soprattutto con dischi fissi particolarmente lenti e partizioni con molti dati.
Al termine dell'operazione riavviare la macchina magari con un dischetto di boot oppure modificando le informazioni del lilo o boot loader che sia.

Nel sito Internet sono presenti le patch sia per il kernel 2.2.x che per il kernel 2.4.x, l'ultima release è la 1.0.10.
E' limitato per il momento a blocchi di 4 KB, non ha ancora una utility di defrag funzionante, non permette di aumentare le partizioni, non supporta le quota ed i log non possono risiedere

Bonnie++

EXT2 - 20 Mb

Sequential Output						*Sequential Input*				*Random Seeks*	
Per Chr		Block		Rewrite		Per Chr		Block			
K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	sec	%CPU
9980,0	86,0	----	---	----	---	8741,0	80,0	----	---	1581,6	3,0

EXT2 - 16 File

Sequential Create						*Random Create*					
Create		Read		Delete		Create		Read		Delete	
/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU
711,0	84,0	----	---	----	---	719,0	79,0	----	---	2393,0	85,0

EXT2 - 20 Mb

Sequential Output						*Sequential Input*				*Random Seeks*	
Per Chr		Block		Rewrite		Per Chr		Block			
K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	sec	%CPU
8876,0	83,0	----	---	----	---	9194,0	83,0	----	---	1567,6	4,0

JFS - 20 Mb

Sequential Output						*Sequential Input*				*Random Seeks*	
Per Chr		Block		Rewrite		Per Chr		Block			
K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	sec	%CPU
4703,0	41,0	----	---	----	---	9940,0	85,0	----	---	1581,0	3,0

JFS - 16 File

Sequential Create						*Random Create*					
Create		Read		Delete		Create		Read		Delete	
/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU
3748,0	16,0	----	---	526,0	2,0	1573,0	16,0	----	---	285,0	2,0

JFS - 20 Mb

Sequential Output						*Sequential Input*				*Random Seeks*	
Per Chr		Block		Rewrite		Per Chr		Block			
K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	sec	%CPU
4703,0	41,0	----	---	----	---	9940,0	85,0	----	---	1581,0	3,0

EXT3 - 16 File

Sequential Create						*Random Create*					
Create		Read		Delete		Create		Read		Delete	
/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU
487,0	82,0	----	---	23643	64	503,0	83,0	----	---	1695,0	79,0

ReiserFS - 20 Mb

Sequential Output						*Sequential Input*				*Random Seeks*	
Per Chr		Block		Rewrite		Per Chr		Block			
K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	sec	%CPU
9206,0	83,0	----	---	----	---	9292,0	76,0	----	---	1573	3,0

ReiserFS - 16 File

Sequential Create						*Random Create*					
Create		Read		Delete		Create		Read		Delete	
/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU
10642	80,0	----	---	11496	84	8850,0	72	----	---	9735,0	84,0

JFS - 16 File

Sequential Create						*Random Create*					
Create		Read		Delete		Create		Read		Delete	
/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU
3748,0	16,0	----	---	526,0	2,0	1573,0	16,0	----	---	285,0	2,0

XFS - 20 Mb

Sequential Output						*Sequential Input*				*Random Seeks*	
Per Chr		Block		Rewrite		Per Chr		Block			
K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	K/sec	%CPU	sec	%CPU
9889,0	85,0	----	---	----	---	9835,0	85,0	----	---	1578,0	3,0

XFS - 16 File

Sequential Create						*Random Create*					
Create		Read		Delete		Create		Read		Delete	
/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU	/sec	%CPU
829,0	20,0	----	---	864,0'	18,0	822,0	21,0	----	---	248,0	5,0

in partizioni dedicate. Nelle distribuzioni SuSE 7.3 e Mandrake 8.1 l'installazione utilizzando il suddetto filesystem porta all'inutilizzo della macchina, in quanto la versione di sviluppo presente nei kernel (1.0.5) non è stabile.

Filesystem XFS

Il filesystem XFS è stato sviluppato dalla Silicon Graphics per il sistema operativo proprietario IRIX e ora è disponibile per piattaforma GNU/Linux con licenza GPL.
Le caratteristiche sono notevoli: 18 mila Petabyte per le dimensioni del filesystem e 9 mila Petabyte per i file.
Supporta l'allocazione dinamica degli i-node e sfrutta gli extent sia per la ricerca dei blocchi liberi (unico per ora), sia per l'organizzazione dei file.
Non è possibile convertire una partizione *ext2* ad *XFS,* tuttavia con un procedimento analogo a quello precedente si arriva al medesimo risultato.
Nel sito Internet sono presenti le patch per il kernel 2.4 e le utility necessarie, attualmente l'ultima release è la 1.0.2. Rispetto allo JFS supporta le quota, e dimensione dei blocchi variabile e fino a 64kb, oltre ad una utility per ridimensionare le partizioni.
La dotazione comprende gli *'xfsprogs'* come pacchetto base.

Filesystem ReiserFS

Il filesystem ReiserFS è stato sviluppato dalla NAMESYS società indipendente e inserito da molto tempo nella distribuzione SuSE.
I limiti del filesystem sono: 16 Terabyte per il filesystem e 4 Gigabyte per i file. La dimensioni dei blocchi è fissata a 4kb.
Supporta l'allocazione dinamica degli i-node mentre non fa uso degli extents previsti comunque per la versione 4 in uscita nel 2002.
Caratteristica di questo filesystem è che invece di usare la struttura a B+Tree usa una versione estesa nominata *B*Tree,* dove ogni oggetto nel filesystem è una struttura *B*Tree.* Il codice sorgente è disponibile nel kernel 2.4. E' presente una utility per ridimensionare le partizioni.

Conclusioni

Un po' di numeri non guastano. Poiché sono appena stati implementati nel laboratorio dell'IPC Fabio Besta non ho ancora dati reali sul campo, ma è possibile con qualche benchmark giudicare indicativamente le prestazioni dei filesystem studiati. Le condizioni di prova sono una workstation "Dell Precision 220" con 256Mb di RAM RIMM, Red Hat Linux 7.1 con tutti gli upgrade e kernel 2.4.14 installata su un HD QUANTUM ATLAS 10K2 (9Gb/10000rpm) pilotato da un controller Adaptec 29160N, il disco per le prove è un QUANTUM VIKING II (4.5Gb/7200 rpm) pilotato da un controller Adaptec 29240UW. La partizione di root è gestita dallo JFS versione 1.0.10.
Per completare la compilazione del kernel senza problemi prima è necessario installare le patch dello XFS, poi dello JFS.
Le prove sullo *ext3* sono state effettuate con lo stesso hardware però con kernel 2.4.9-12. I software utilizzati sono *'bonnie++'* e *'postmark'.*
Il primo simula il comportmanento in presenza di accessi a file singoli e di piccole dimensioni. L'utilizzo avviene per input in linea di comando. Per completare i test si sono eseguiti le seguenti procedure:

```
[root@localhost /root]#
 bonnie++ -d /mnt/rootfs
           -s 20 -r 4 -u 0
         >bonnie.ext2.20M.csv
```

Il secondo è più adatto per testare il filesystem in presenza di applicazioni squisitamente Internet orientate: come un server di posta. L'utilizzo avviene con *'bash'* integrata:

```
[root@localhost /root]#
                  postmark
Postmark v1.5: 3/27/01
pm>set number 1000
pm>set transactions
10000
pm>run
```

Di seguito sono riportati i risultati delle prove.

Umberto Zanatta

Postmark

EXT2	
Time:	2 seconds total
	2 seconds of transactions *(5000 per second)*
Files:	5994 created *(2997 per second)*
	4999 read *(2499 per second)*
	4968 appended *(2484 per second)*
	5994 deleted *(2997 per second)*
Data:	31.93 Mb read *(15.97 Mb per second)*
	37.88 Mb written *(18.94 Mb per second)*

EXT3	
Time:	11 seconds total
	7 seconds of transactions *(1428 per second)*
Files:	5994 created *(544 per second)*
	4999 read *(714 per second)*
	4968 appended *(709 per second)*
	5994 deleted *(544 per second)*
Data:	31.93 Mb read *(2.90 Mb per second)*
	37.88 Mb written *(3.44 Mb per second)*

ReiserFS	
Time:	9 seconds total
	8 seconds of transactions *(1250 per second)*
Files:	5994 created *(666 per second)*
	4999 read *(624 per second)*
	4968 appended *(621 per second)*
	5994 deleted *(666 per second)*
Data:	31.93 Mb read *(3.55 Mb per second)*
	37.88 Mb written *(4.21 Mb per second)*

JFS	
Time:	127 seconds total
	106 seconds of transactions *(94 per second)*
Files:	5994 created *(47 per second)*
	4999 read *(47 per second)*
	4968 appended *(46 per second)*
	5994 deleted *(47 per second)*
Data:	31.93 Mb read *(257.47 kb per second)*
	37.88 Mb written *(305.46 kb per second)*

XFS	
Time:	19 seconds total
	16 seconds of transactions *(625 per second)*
Files:	5994 created *(315 per second)*
	4999 read *(312 per second)*
	4968 appended *(310 per second)*
	5994 deleted *(315 per second)*
Data:	31.93 Mb read *(1.68 Mb per second)*
	37.88 Mb written *(1.99 Mb per second)*
Links:	http://www-124.ibm.com/developerworks/oss/jfs
	http://oss.sgi.com/projects/xfs
	http://www.namesys.com

SATAN - Il primo ed il più famoso scanner di rete

SATAN, scritto da Wietse Venema e Dan Farmer è l'acronimo di Security Administrator Tool for Analyzing Networks, scopriamone insieme le potenzialità..

Ecco la frase estratta da USENET, postata l'8 Marzo 1995, che fece diventare famoso questo tool: *"SATAN was written because we realized that computer systems are becoming more and more dependent on the network, and at the same becoming more and more vulnerable to attack via that same network."*. Come effettivamente ci ricorda questa frase, già da allora, ma molto di piu' ultimamente, i computer sono dipendenti in rete e sono vulnerabili ad attacchi nella stessa rete.
Accedere in un computer, significa nella maggioranza dei casi, accedere a tutte le risorse della rete a cui la macchina "bucata" è interconnessa.
SATAN, nacque quindi come tool per aiutare gli amministratori di sistema a gestire il problema dell'eventuale intrusione di qualcuno nel computer. I problemi che furono fatti analizzare dal SATAN, sono questi:

- File system NFS esportabile a qualunque host.
- File system NFS esportato a programmi senza privilegi dovuti.
- File system NFS esportato via portmapper (sunrpc 111/tcp portmapper ; sunrpc 111/udp portmapper).
- File delle password NIS accessibile da qualunque host.
- Vecchie versioni buggate di sendmail, precedenti alla 8.6.10.
- Accesso tramite REXD da host arbitrari.
- Controllo degli accessi al server X disabilitato.
- Qualunque file accessibile via TFTP *(tftp 69/tcp; tftp 69/udp)*.
- Shell remota accessibile tramite qualunque host (usando rlogin).
- Home directories scrivibili usando FTP con account anonimo.

Questi problemi, sono molto vecchi infatti il tutto risale ad advisories del CERT datati inizi anni '90, ovvero il tool risultava praticamente inutile. Perché allora si è scelto di scrivere un articolo su qualcosa di datato? E' vero che SATAN fu usato da molti sysadm come supporto "anti-intrusione", ma é pur vero che gli utilizzatori principali di tale tool, sono stati gli hackers (direi crackers...). Si è vista quindi una grande espansione di questi tool, dal semplice port scanner al più utile vulnerability scanner! Ma SATAN, è andato perso?!? Chiaramente no, ne abbiamo un ottimo successore: il SAINT. Questo tool è rilasciato sotto licenza SATAN, poiché deriva dal lavoro svolto da Wietse Venema e Dan Farmer. SAINT si scarica dal sito: *http://www. wwdsi.com/ saint/*. Il tool è nato come un'implementazione avanzata del SATAN, però essendo opensource, fa parte di una catena di prodotti commerciali della World Wide Digital Security Inc. *(wwdsi)*, gli stessi, sono stati creati per tutti coloro che hanno necessità di strumenti in grado di produrre log e report personalizzati in base alle esigenze o creare report web. SAINT, come SATAN è quindi scritto in PERL e funziona sulle seguenti piattaforme:

- Linux (x86)
- SunOS (sparc)

- HP-UX (HPPA)
- OpenBSD (x86)
- FreeBSD (x86)
- NetBSD (x86)

Altra derivazione del SATAN è SARA *(http://www-arc.com)*.

Come funziona SATAN?

I vulnerabilities scanner, controllano gli host che vengono passati come parametro, procedendo innanzitutto alla risoluzione dell'host in ip, controllando se il portmapper è attivo *(sunrpc 111/ tcp)* ed in caso affermativo, i servizi RPC che sono stati mappati. Finiti questi controlli, viene effettuato un port scanner, per sapere che servizi sono attivi, inoltre i vulnerabilities scanner, verificano la versione dei demoni installati sulla macchina, per esempio se è attiva la porta 21, ossia FTP, viene controllato il banner iniziale:

```
[root@r0Ot root]# telnet
localhost 21
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 r0Ot FTP server
    (Version wu-2.6.1-18)
            ready.
```

In questo particolare caso si è effettuata la prova "manuale" e abbiamo verificato che l'FTP server attivo sulla macchina locale, nel nostro caso Wu-Ftpd 2.6.1-18, è vulnerabile *(Security Focus Vulnerability 3581)*. Se SAINT o SARA, hanno quindi installato un plugin in grado di riconoscere tale vulnerabilità, questa verrà segnalata nei report in modo da essere consci del fatto che bisogna aggiornare il demone. Altra notevole funzionalità dei vulnerabilities scanner (non di tutti però), è il fatto di poter "indovinare" il sistema operativo presente sulla macchina remota controllando il fingerprint della macchina.

SANS, cos'è?

SANS è l'acronimo di System Administration, Networking, and Security, un'istituto fondato nel 1989 e comprende 96000 persone fra amministratori di sistema, esperti di sicurezza ed amministratori di rete. Lo scopo di questa organizzazione è quello di condividere tutte le soluzioni inerenti problemi di sicurezza, di rete e di amministrazione, che altri amministratori hanno risolto in maniera efficace e brillante. L'istituto ha certificato i due scanner principali che verranno di seguito trattati ossia SAINT e SARA.

SAINT: L'installazione

Per procedere con l'installazione di Saint, dobbiamo accertarci che il nostro sistema abbia almeno i seguenti requisiti di sistema:

- Un Sistema Operativo Unix-like: Linux, FreeBSD, OpenBSD, NetBS, Solaris, AIX, HP-UX, IRIX, ecc...
- PERL di versione uguale o superiore alla 5.004 (Versione stabile al momento 5.6.1).
- Un Web Browser: Netscape, Opera, Lynx, Links, Chimera, ecc..
- Una macchina che abbia abbastanza risorse, poichè SAINT ne richiede davvero tante per effettuare degli scan su larga scala!

Requisiti opzionali ma raccomandati

SAINT senza questi due tool funziona comunque, consigliamo tuttavia l'installazione degli stessi:

- **NMap** -> Viene utilizzato da SAINT per identificare il sistema operativo che stiamo controllando (fingerprint).
- **Samba Tool** -> Questi tool sono usati per determinare durante scan su macchine windows, il nome della macchina (netbios name) e gli share pubblici della macchina (condivisioni).

L'installazione risulta molto semplice, veloce e pratica:

```
./configure
make
make install
```

Il primo comando crea il Makefile a seconda del vostro sistema operativo ed a seconda di come lo stesso è preconfigurato. Il secondo comando provvede a compilare il SAINT in modo che funzioni sulla Linux Box attiva: il terzo installa il SAINT. Per eseguire Saint basta digitare:

```
./saint
```

In caso di configurazioni particolari, ad esempio state dietro un firewall, vi rimandiamo alla guida contenuta nel README. L'aiuto per il SAINT in modalità interattiva si trova nel link visualizzato dal browser durante l'esecuzione dello stesso. Per modificare il browser per la visualizzazione dei dati, in caso il netscape di default non ci vada bene, basta modificare la variabile *$MOSAIC,* che troveremo definita nel file *config/paths.pl,* con il programma completo di *PATH* che vogliamo utilizzare, ad esempio:

```
$MOSAIC="/usr/bin/links";
```

Installiamo SARA

Come nel caso di SAINT, i requisiti di sistema di SARA, sono la presenza del PERL con versione uguale o superiore alla 5.0.3, un web browser e chiaramente un sistema unix-like! Nel Makefile troviamo come target supportati, i seguenti:

```
aix osf bsd bsdi dgux irix4
irix freebsd
        hpux9 hpux linux
```

linux-nansi sunos4 sunos5
trusted-sunos5 sysv4

L'installazione è semplicissima, basta impartire il comando:

```
make linux
```

(Poichè nel nostro caso installiamo il SARA su un sistema Linux Redhat 7.2) Installati i due tool, passiamo nell'effettuare alcune prove per testarne l'efficacia!

L'INTERFACCIA

L'interfaccia principale del SATAN si presenta come in Figura 1. Il Data Management è la sezione che gestisce la creazione, l'apertura e la modifica dei suoi database. I database sono rispettivamente 3:

1. facts
2. all-hosts
3. todo

Il database "facts", contiene le informazioni sui servizi e vulnerabilità del sistema che è stato analizzato, i campi del database sono:

- **Target ->** Nome dell'host di riferimento del record.
- **Service ->** Il nome del servizio testato.
- **Status ->** Stato dell'host, ad esempio se raggiunto o meno.
- **Severity ->** La gravità della vulnerabilità del servizio testato.
- **Trusted ->** L'utente che ha dato il permesso di far accedere da remoto la macchina.
- **Trustee ->** Segnala gli host che possono accedere da remoto con privilegi dell'utente trusted.
- **Canonical Service Output ->** Se il servizio non è vulnerabile restituisce i parametri che il demone ha dato in uscita, se vulnerabile definisce il tipo di vulnerabilità.
- **Text ->** Campo per inserire commenti sul report.

Il database *"all-hosts"* tiene traccia di tutti gli host analizzati da SATAN, inclusi anche quelli non esistenti. Il database "*todo*" segna quello che bisogna fare per risolvere i problemi di sicurezza. La sezione *Target Selection* serve per selezionare l'host o gli hosts che SATAN deve analizzare e consente anche di definire il livello di profondità dell'analisi:

1 Analisi Leggera.

2 Analisi Media (questo tipo di analisi viene loggata).

3 Analisi Pesante (questo tipo di analisi viene pesantemente loggata ed appaiono messaggi all'amministratore nella system console).

La sezione *Reporting & Data Analysis* ci consente di leggere i risultati delle analisi che il SATAN ha fatto:

- Vulnerabilità.
- Informazioni sull'host analizzato.
- Hosts che hanno accesso alla macchina (rlogin, NFS shares).

Infine troviamo il *SATAN Configuration Management,* ovvero il pannello di controllo di SATAN, che permette di configurarlo, usando le impostazioni ritenute più opportune.
SARA E SAINT, hanno più o meno la stessa interfaccia, chiaramente differiscono da SATAN per le funzionalità che sono state aggiunte e per la modularità (plug-ins).
In definitiva si rivelano degli ottimi tool per controllare la sicurezza delle nostre reti.

Francesco Marasco

Figura 2

I diversi livelli di scanning.

Figura 1

Il Control Panel di SATAN.

Con questo articolo inizia il corso di JSP, il cui scopo non è certo quello di formare programmatori esperti in questo particolare ambiente di sviluppo, ma di fornire una panoramica quanto più completa ed esaustiva possibile di questa piattaforma.

Java Server Pages primi passi...

A nostro avviso Java Server Pages è attualmente il più avanzato e innovativo ambiente di sviluppo, nel panorama dei linguaggi per lo sviluppo di applicazioni web.

Premessa

Anche se la diffusione di questo linguaggio non è attualmente al pari di altri come il PHP, crediamo che i tempi siano maturi per un suo utilizzo concreto nello sviluppo di applicazioni reali in qualsiasi ambito offerto dal web, infatti se fino a qualche tempo fa, c'era qualche diffidenza, dovuta forse a gap prestazionali in termini velocistici, delle applicazioni server side basate su java, ora i maggiori produttori di piattaforme per lo sviluppo di applicazioni per il web come IBM e Oracle, fanno di JSP e della piattaforma J2EE (*Java 2 Enterprise Edition*) il loro prodotto di punta contro i diretti concorrenti come la suite PHP-Apache, troppo relegata alla sola architettura Linux e allo sviluppo di piccole e poco manutenibili applicazioni per uso quasi personale, e l'architettura .NET di Microsoft di cui Ximian ha recentemente annunciato il porting per Linux, attualmente ancora in fase quasi embrionale. Scrivere applicazioni in JSP richiede una buona conoscenza di JAVA, e delle problematiche relative allo sviluppo di applicazioni multithread oltre a una concreta fase di progettazione antecedente la stesura del codice e una certa macchinosità delle procedure di sviluppo delle applicazioni stesse. I vantaggi però sono notevoli e derivanti sopratutto dalla sua diretta discendenza dal java, a cominciare dalla portabilità e dalla scalabilità offerta dalle applicazioni scritte in questo linguaggio. Faremo conoscenza di oggetti come *Servlet* e *Java Beans* che costitiuscono assieme alle JSP i cardini della piattaforma J2EE e del modello MVC *(Model View Controller)* di Sun che auspica la separazione dei dati (il modello), rappresentati dagli Entity Beans, dalla loro interfaccia di accesso *(View)*, realizzata per esempio, ma non solo, via web tramite pagine JSP, e dalle operazioni che possono essere svolte su di essi gestite tramite Servlet o Session Bean (che svolgono la funzione di Controller) Tutti e tre questi componenti

Figura 1

L'Home Page che si presenta dopo aver installato Tomcat.

naturalmente possono avvantaggiarsi di tutte le tecnologie java come *RMI, CORBA, JNDI,* ecc. In realtà per lo sviluppo di piccole applicazioni non è necessario attenersi strettamente al modello MVC che in questo caso risulterebbe pesante e poco gestibile da una sola persona e si confà a realtà aziendali ben più grandi. Per cominciare iniziamo con i file che dobbiamo procurarci per creare il nostro ambiente di sviluppo JSP. Il contenitore prescelto è il Tomcat del progetto Jakarta di Apache, scritto in java, giunto alla versione 3.3,offre il supporto ufficiale alle *ServletApi 2.2* e *JSP 1.1* e rappresenta per noi la scelta più adeguata in quanto è il contenitore open source consigliato da Sun ed è facilmente integrabile con Apache, vedremo in seguito come, per ora lo utilizzeremo solo come server JSP stand-alone. Dal sito della Sun che offre il supporto ai programmatori java *(http://java.sun.com)* scarichiamo il *jdk1.3.1,* per Linux, dal sito del progetto Jakarta *(http://jakarta.apache.org/tomcat/)* scarichiamo il Tomcat ver 3.3, in formato binario cioè già compilato e compresso in formato *tar.gz* e le *jakarta-Servlet-api* che serviranno per sviluppare i bean e le Servlet. Dopo aver installato il jdk nel formato che vi è piu' familiare per esempio nella directory */usr/src/jdk1.3.1* installiamo il Tomcat nella directory */usr/src/jakarta-tomcat-3.3* e le Servlet api nella directory */usr/src/jakarta-Servletapi-2.2,* aggiungiamo al file */etc/profile* le seguenti variabili d'ambiente:

```
TOMCAT_HOME=/usr/src/
          jakarta-tomcat-3.3
JAVA_HOME=/usr/src/jdk1.
                        3.1
CLASSPATH=$CLASSPATH:.:
                  /usr/src/
   jakarta-Servletapi-2.2/src
export TOMCAT_HOME
    JAVA_HOME CLASSPATH
```

Bene a questo punto siamo pronti per far partire il nostro Tomcat con il comando */usr/src/jakarta-tomcat-3.3/bin/startup.sh,* mentre per terminare l'esecuzione useremo il comando */usr/src/jakarta-tomcat-3.3/bin/shutdown.sh.* Passiamo alla descrizione della struttura delle directory del Tomcat:

- la directory *bin* contiene gli script principali che consentono per esempio l'avviamento e la chiusura del server;
- la directory *conf* contiene i file di configurazione che per ora lasceremo inalterati;
- la directory *Webapps* contiene le applicazioni web.

Un'applicazione Web secondo Tomcat e secondo le specifiche Sun è un'insieme di pagine html, JSP, immagini, classi java e quant'altro serva al nostro sito organizzato nella seguente struttura a directory:

```
$TOMCAT_HOME-----*
        |
   /webapps----*
        |
     /myapp--*
        |
   /WEB-INF---*
        |
    /classes/
        |
      / lib/
```

La directory *myapp* posizionata nella directory *webapps* di tomcat contiene la mia applicazione web, nella sua radice sono contenute le pagine html, JSP e le directory di supporto per contenere immagini ecc. Le Servlet e i bean in formato *jar* sono posizionati in *myapp/WEB-INF/lib* mentre quelli in formato *.class* nella directory */myapp/WEB-INF/classes* e da questa directory si diramano gli eventuali packages utilizzati. Per scrivere la nostra prima applicazione dobbiamo creare una struttura a directory come questa a partire dalla directory *WebApps* di Tomcat che la includerà nei propri contesti a partire dal prossimo riavvio. In realtà la struttura di un'applicazione Web è più complessa,prevede l'inclusione di speciali file di configurazione e di altre directory particolari che vedremo in seguito, quella proposta è solo la struttura base sufficiente per cominciare e completamente funzionante.

Prima pagina JSP

Ora creiamo la nostra prima pagina JSP chiamiamola *index.JSP* e posizioniamola nella nostra directory *myapp,* il sorgente è questo:

```
<%@ page language="java"
contentType="text/html" %>
<html>
<head><title>JSP first
          page</title></head>
 <body>
<% out.print("<h2>Ciao
          Mondo</h2>"); %>
 </body>
</html>
```

A questo punto digitiamo nel nostro browser l'url *http://localhost:8080/myapp/* e il gioco è fatto; vedremo apparire il messaggio *'Ciao Mondo'* nel nostro browser.
Diamo ora un'occhiata agli elementi sintattici delle pagine JSP.

Istruzioni *<%@ ...%>*
Gli elementi racchiusi tra questi tag sono le istruzioni JSP, abbiamo già incontrato l'istruzione *page,* altre sono *include* e *taglib,* avremo occasione di scoprire in dettaglio il loro utilizzo nelle prossime puntate, per ora basti sapere che questi elementi sintattici influenzano tutto il contenuto di una pagina.

Tag Standard
In JSP esistono tag proprietari da utilizzare nelle pagine oltre a quelli HTML, esiste la possibilità di creare proprie librerie di tag o di importarne altre sviluppate da terzi, esempi sono:

```
<JSP:usebean    />,
          <JSP:forward.../>,
            <JSP:pluggin..../>
```

Scriptlet
Sono racchiusi tra i tag *<%, %>* ,rappresentano gli elementi di scripting

in linguaggio java, possono trovarsi in qualsiasi punto della pagina. L'istruzione:

```
<% out.print("<h2>Ciao
        Mondo</h2>"); %>
```

della nostra piccola pagina di esordio ne è un esempio.

Dichiarazioni
Racchiuse tra i Tag *<%!....%>* servono per dichiarare variabili di istanza o metodi della classe *java* corrispondente alla pagina JSP.

Commenti
Sono compresi tra i tag *<%-- ... --%>*

Nei prossimi articoli avremo modo di conoscere e usare tutti questi elementi e i diversi aspetti della programmazione che li riguardano. Approfondiamo ora due aspetti fondamentali della programmazione JSP, i *Java Beans* e le *Servlet*. I *bean* sono istanze di classi java, quindi oggetti, che una volta istanziati da una pagina JSP o da una Servlet hanno se necessario, la capacità di restare in attesa dell'invocazione di un loro metodo all'interno del contenitore JSP che stiamo usando, anche dopo la fine dell'esecuzione della pagina che li ha istanziati, in modo da poter essere utilizzati da altre pagine e da altre istanze della stessa, possono essere serializzati e mantenere il loro stato anche dopo lo shutdown del server. La piattaforma J2EE introduce anche gli *Enterprise Java Beans* distinguendo tra *Entity Bean* e *Session Bean* che dovrebbero rappresentare rispettivamente un'astrazione dei dati su cui stiamo operando e l'entità delegata a svolgere azione su di essi. Dal punto di vista strutturale i *Java Bean* (non gli *EJB*) sono delle normali classi java che implementano l'interfaccia standard *Serializable* e offrono una serie di metodi di scrittura dei dati chiamati *setter,* perchè iniziano con il prefisso *'set'* e metodi di lettura dei dati chiamati *ghetter* perchè iniziano con il prefisso *'get',* se il tipo del dato in questione è booleano il suo metodo *getter* avrà prefisso *is* invece che *get.* Diciamo che queste regole non sono solo convezioni sintattiche, ma sono necessarie, infatti le proprietà dei bean visibili in una pagina JSP sono accessibili sia in lettura che in scrittura tramite i tag personalizzati *<JSP:usebean../>* *<JSP:getProperty ../>* e *<JSP:setProperty../>,* il cui utilizzo vedremo presto. Questi non fanno alcun riferimento ai metodi del bean stesso, ma solo al nome della proprietà a cui permettono di accedere ipotizzando che per essa esista un metodo *ghetter* e un metodo *setter* che rispetti le regole sopra indicate. Le *Servlet* invece rappresentano il primo tentativo di Sun di portare la tecnologia java nello sviluppo di applicazioni orientate al web. Sono oggetti raggiungibili via web attraverso un'url, hai quali una volta passati dei parametri tramite i normali metodi *GET* o *POST,* si ottiene l'invocazione di determinati metodi sul server e si riceve in risposta una pagina html generata dinamicamente esattamente come una volta si usavano gli script CGI scritti in perl, ma avendo a disposizione tutte la API del java. La loro importanza risiede nel fatto che ogni pagina JSP viene trasformata dal server in una servlet e compilata in bytecode java prima di essere eseguita. In questo senso pagine JSP e Servlet sono interamente interscambiabili, le pagine JSP rappresentano solo un escamotage sintattico, ma non semantico, per facilitare la realizzazione di applicazioni. Infatti una pagina JSP può essere realizzata addirittura senza l'utilizzo di scriptlet al suo interno, lasciando svolgere le operazioni sui dati a tag personalizzati, realizzati da terzi, e quindi senza conoscere neppure una parola di java. Un'altra prova dello stretto legame che esiste tra pagine JSP e Servlet è che questi componenti possono condividere oggetti (in particolare Java Beans) dichiarati nell'uno o nell'altro contesto; questo rende la loro integrazione e il loro utilizzo comune, più semplice e produttivo nell'ottica dello sviluppo di un progetto che sia più modulare e scalabile possibile.

Figura 2
Pagina di prova di un nostro progetto JSP.

Conclusioni

Bene, con questo termina la nostra breve panoramica sul mondo delle Java Server Pages, dopo questa necessaria e speriamo esaustiva introduzione, dal prossimo numero cominceremo a proporre esempi concreti e sempre più interessanti.

Antonio Pace
Giuseppe Tradigo

A lezione con il PostgreSQL

In molte scuole italiane si insegna in ambito curriculare la gestione delle basi di dati. Il riferimento è alle classi quarte e quinte degli Istituti professionali per il commercio (indirizzo informatico) e degli Istituti tecnici per programmatori, siano essi industriali che commerciali.

Negli Istituti professionali ad indirizzo informatico, contrariamente al buon senso, l'insegnamento della disciplina informatica, non prevede l'ausilio del laboratorio di informatica (al riguardo si veda *Fulvio Ferroni: "Il software libero nella didattica (II)" in Linux Magazine n. 12 giugno-luglio 2001, http://linuxdidattica.org.*
Privare l'insegnante di informatica del rispettivo laboratorio corrisponde a privare un insegnante di un qualsivoglia libro di testo o strumento didattico e in ultima istanza, oggi, significa privarlo dell'esercizio della libertà d'insegnamento.
Nella scuola Fabio Besta di Treviso (Istituto Professionale di Stato ad indirizzo informatico), grazie al software libero e all'autonomia scolastica (ben interpretata dagli Organi dirigenti della scuola), tale assurdità è stata rimossa riuscendo così a far provare agli studenti, quello che si insegna in teoria. In questo modo ogni studente può rendersi conto anche praticamente delle varie problematiche inerenti la gestione di una base di dati e di un sistema informativo più in generale.
La soluzione al paradosso: l'utilizzo di PostgreSQL in ambiente GNU/Linux.
Utilizzando un vecchio elaboratore Pentium, venticinque i286 e una stampante, tutte macchine da rottamare, è stata costituita una rete TCP/IP basata sul sistema operativo GNU/Linux installato sul Pentium (rete 192.168.1.0). Durante l'installazione del sistema operativo è stato installato anche il DBMS PostgreSQL 7.0.2 e il suo client, a interfaccia a carattere, psql.
Sulle macchine i286 si è installato il Dos Caldera e un programma telnet per Dos. Tutte le macchine i286 sono state collegate al Pentium tramite due HUB.
Nella macchina GNU/Linux oltre al binario e ai sorgenti di PostgreSQL e di psql si è anche installata tutta la relativa documentazione nella directory: */usr/share/doc/postgresql-7.0.2/.*
Per ogni studente, su GNU/Linux, è stata creata un'utenza (account). Si è, successivamente, configurato, con i permessi dell'utente root, il file *pg_hba.conf* per dare agli utenti gli accessi, senza restrizioni *(trust),* al server PostgreSQL:

```
#
#Example PostgreSQL
   host access control file.
#
......
host all 127.0.0.1
   255.255.255.255 trust
```

Per maggiori dettagli sull'installazione e configurazione di PostgreSQL si veda *Umberto Zanatta: "La rete: Linux per la didattica" cap. 13, in http//linuxdidattica.org).*
Per ogni utente del sistema GNU/Linux si crea un utente PostgreSQL con il comando *createuser.*
Si farà in modo, per semplicità, che siano uguali *(utente linux=utente PostgreSQL):*

```
toni$createuser toni
                    <invio>
Shall the new user be
  allowed to create databa-
        ses? (y/n) n <invio>

Shall the new user be
   allowed to create more
      new users? (y/n) n
                    <invio>
CREATE USER
```

Ogni studente si collega, dalla macchina i286, a GNU/Linux digitando:

```
C>telnet 192.168.1.1
                    <invio>
```

```
login: toni <invio>
passwd:       <invio>
```

apparirà il prompt (il Pentium si chiama *cea*):

```
toni@cea$
```

E qui siamo in ambiente GNU/Linux.
Dopo aver progettato in classe e analizzato la base di dati biblioteca (modello E-R, schema logico, schema fisico, ecc.) si passa all'esercitazione. Tale base di dati e la relativa esercitazione sono documentate in *Daniele Giacomini "Appunti di informatica libera, Le funzioni e i trigger in PostgreSQL: un'esercitazione didattica", http://linuxdidattica.org.*
Per prima cosa si parte dalla creazione della base di dati da parte dell'amministratore postgres. L'utente *toni* si trasforma nell'utente *postgres*:

```
toni@cea$su - postgres
                         <invio>
bash-2.04$createdb
            biblioteca <invio>

CREATE DATABASE
```

usciamo dall'utente postgres con il comando

```
exit
bash-2.04#exit <invio>
toni@cea$
```

Gli studenti, già a livello di progettazione della base di dati in classe, vengono suddivisi in gruppi dove a ogni gruppo è assegnato un compito e una gestione ben precisa di una parte della della base di dati.
Anche se utilizziamo solo una macchina GNU/Linux le esercitazioni sono obbligatoriamente in rete, in ambiente multiutente, client-server dove la concorrenza di accesso e la gestione delle transazioni si può verificare pienamente. Ne risulta una esercitazione più vicina alla realtà aziendale rispetto alle solite esercitazioni monoutente.
La base di dati *biblioteca*, serve per gestire i prestiti dei libri di una biblioteca, ed è formata dalle cinque tabelle riportate in Figura 1.
I singoli gruppi di studenti creano le tabelle (istruzione *create*) e le vanno a popolare (istruzione *insert into*).

```
libri(n_inv,autore,titolo,collocazione,soggetto,cod_ed,
                                          prezzo,an_ed)
utenti(cod_ut,nome,cognome,telefono,indirizzo,citta)
localita(citta,cap,prov,nz)
editori(cod_ed,rag_soc,indirizzo,citta,telefono)
prestiti(n_prog,n_inv,cod_ut,data_p,data_r)
```

Figura 1

Schema del database biblioteca (grassetto= chiave primaria, corsivo=chiave esterna).

Tutti i gruppi devono popolare la tabella *prestiti*.
In questa esercitazione si utilizza la capacità del PostgreSQL e del psql di elaborare file ASCII dove sono scritte istruzioni SQL. Gli studenti scrivono le istruzioni in un file con l'editor *VI,* per poi eseguirlo all'interno della base di dati con il metacomando *\i <nomefile>*, come vedremo più avanti.
Ovviamente, tutti i gruppi dovranno dare i permessi agli altri studenti (o toglierli, a seconda dei casi), dalle tabelle che creano, tramite le istruzioni *grant* e *revoke.*

Al lavoro con il server PostgreSQL e il client psql

Cominciamo con un esempio immaginando di essere l'utente *toni* che deve creare e popolare *prestiti*:

```
C>telnet 192.168.1.1
                       <invio>
login:toni <invio>
passwd:      <invio>
toni@cea$psql -h localhost
     -d biblioteca -U toni
                       <invio>
biblioteca=>
```

Ora siamo connessi al database *biblioteca.*
A questo punto l'utente toni può utilizzare i metacomandi psql o le istruzioni SQL. Psql è il client a interfaccia a carattere.
I metacomadi psql, che permettono anche di interfacciarsi con GNU/Linux, iniziano sempre con la barra rovesciata \, come ad esempio: *\q* che serve per uscire, *\d* per vedere le tabelle della base di dati, *\l* per vedere le varie basi di dati, *\d <nometabella>* per vedere lo schema della tabella, *\?* la guida di psql, *\h* per avere la guida SQL, *\!* per utilizzare i comandi del sistema operativo, ecc.
L'utente *toni* crea la tabella *prestiti* utilizzando l'editor *VI* come mostrato in Figura 2.
Usciamo dal *VI* digitando i consueti *esc:wq* e mandiamo in esecuzione il file *creo_prestiti.sql:*

```
biblioteca=>\i
    creo_prestiti.sql <invio>

CREATE
```

```
biblioteca=>\! vi creo_prestiti.sql
create table prestiti (
n_prog serial,
n_inv char(5) references libri,
cod_ut char(5) references utenti,
data_p date check (data_p<=data_r),
data_r date default '2050.1.1.'
);
```

Figura 2

Creazione della tabella prestiti.

Se non fossero già state create, dagli altri gruppi, le tabelle *utenti* e *libri* l'utente *toni* verrebbe avvisato che non può creare la tabella prestiti (perché fa riferimento alle chiavi primarie *n_inv* e *cod_ut*). Adesso inseriamo, interattivamente, un prestito:

```
biblioteca=>insert into
     prestiti (n_inv, cod_ut,
              data_p) <invio>
biblioteca->values ('12',
'15','2001.10.10');
                       <invio>
INSERT 1
```

Nell'inserimento dei dati subito si incontra il problema dell'integrità referenziale esterna.
Non si può popolare una tabella all'interno della quale un campo fa riferimento a una chiave primaria inesistente nell'altra tabella (nel caso precedente deve esistere il libro con n_inv =12 o utente con cod_ut =15).
Quando un libro rientra sarà sufficiente eseguire l'istruzione update, e nell'ipotesi che il numero progressivo del prestito sia 1:

```
biblioteca=>update libri
     set data_r=date('now')
                        <invio>
biblioteca=>where
          n_prog=1; <invio>
UPDATE 1
```

```
toni$vi controlla_inserimento.plpgsql
create function inserisci_prestito_trigger()
returns opaque
  as 'declare
   numero_inventario char(5);
   data_restituzione date;
   prestito record;
    begin
     numero_inventario:=new.n_inv;
     data_restituzione:=''2050.1.1'';
      select into prestito *
      from prestiti
      where n_inv=numero_inventario and
      data_r=data_restituzione;
       if found
        then
          raise exception \'il libro è già a prestito\';
        else
          return new;
        end if;
    end;'
language 'plpgsql';

create trigger controlla_libro_uscito
before insert
on prestiti
for each row
execute procedure inserisci_prestito_trigger();
```

Figura 3

Creazione del file controlla_inserimento.plpgsql.

In una esercitazione didattica è difficile controllare i dati prima che essi vengano inseriti.
Tutti gli studenti dovevano preparare una decina di prestiti a piacere (ricavando i dati dalle tabelle *libri* e *utenti*) precedentemente stampate. Dovevano scrivere questi inserimenti in un file, che successivamente avrebbero eseguito con il metacomando *\i*. In questo modo, tra l'altro, si impara a lavorare anche off line, e non solo in maniera interattiva (alcuni studenti il file se lo sono scritto a casa).
Una volta popolata *prestiti*, osservandola, ci si è accorti che era stato dato a prestito lo stesso libro a più utenti contemporaneamente, cosa evidentemente assurda, come nell'esempio sottostante:

```
biblioteca=>select * from
  prestiti where n_inv='5';
                       <invio>

n_prog, n_inv, cod_ut,
        data_p,      data_r

1       5       12
  2001.10.10   2050.1.1

10      5       25
  2001.11.11   2050.1.1

12      5       34
  2001.11.28   2050.1.1
```

La soluzione è stata trovata utilizzando la caratteristica del PostgreSQL di gestire i trigger e le funzioni in vari linguaggi, tra cui il Plpgsql.
Si procede, perciò, prima ad associare questo linguaggio alla base di dati (operazione da fare come utente postgres) con il comando *createlang:*

```
toni@cea$su - postgres
                       <invio>

bash-2.04$createlang -h
   localhost -d biblioteca --
       pglib=/usr/lib/pgsql/
              plpgsql <invio>
toni@cea$
```

Successivamente come utente *toni* abbiamo creato il file *controlla_inserimento.plpgsql,* come descritto in Figura 3, contenente il trigger e la relativa funzione, precedentemente spiegati in classe.
Questo file è nella directory dell'utente *toni.* Ci colleghiamo al database *biblioteca:*

```
toni@cea$psql -h localhost
       -d biblioteca -U toni
                       <invio>
biblioteca=>
```

A questo punto possiamo compilare il trigger con il metacomando *psql \i:*

```
biblioteca=>\i controlla_
        inserimento.plpgsql
                       <invio>
CREATE
CREATE
```

E' stata creata prima la funzione e poi il trigger. *Cancelliamo* il contenuto del file *prestiti,* precedentemente popolato:

```
biblioteca=>delete from
          prestiti; <invio>
```

```
delete 74
```

Inseriamo dei prestiti eseguendo il contenuto del file *inserimento_prestiti.sql* già predisposto dall'utente *mario:*

```
biblioteca=>\i ..\mario\
    inserimento_prestiti.sql
                    <invio>
INSERT 1

...

INSERT 10
```

Se si ripete l'operazione, si ottiene un errore:

```
biblioteca=>\i ..\mario\
    inserimento_prestiti.sql
                    <invio>

ERROR il libro è già a
                    prestito
.....

ERROR il libro è già a
                    prestito
```

Come si vede (ERROR ecc.) il trigger ha impedito l'inserimento dei libri già a prestito.
Una volta costruita la base di dati biblioteca integra, senza dati contraddittori, e dopo aver fatto rientare alcuni libri con date di rientro a piacere (controllate comunque dalla verifica *check data_p<=data_r*), si è passati ad interrogare i dati per vedere, ad esempio, quali libri erano ancora a prestito oltre il limite di 30 giorni dalla data del prestito, vedi Figura 4.
Gli studenti hanno potuto interrogare la base di dati con altre istruzioni *select* di vario tipo, tutte rivolte a trasformare i dati contenuti nella base di dati in informazioni utili alla gestione della biblioteca. E così hanno potuto verificare sul campo il funzionamento di un semplice sistema informativo.

Conclusioni

All'Istituto Besta di Treviso è stato risolto abbastanza agevolmente il paradosso nel quale la politica del Ministero ha relegato l'insegnamento dell'informatica in questo ordine di scuole. Si è permesso così all'insegnante d'informatica di esercitare appieno il suo diritto alla libertà d'insegnamento potendo egli prima proporre e poi scegliere liberamente il suo libro di testo. E solo grazie al software libero un insegnante di informatica può esercitare il diritto alla libertà d'insegnamento come libera scelta del suo libro di testo.
Ma una domanda sorge spontanea: perché allora il software libero non è così diffuso nella scuola?
La risposta a mio avviso è altrettanto semplice.
Non è solo per mancanza di una politica del Ministero, almeno fino ad oggi, "in soggezione" al monopolio dell'informatica, perché l'insegnante, a difesa del suo lavoro, può sempre avvalersi dell'art. 33 della Costituzione e dell'autonomia progettuale.
Non è solo per mancanza di capacità tecnico-professionali tra i docenti (e operatori scolastici in genere) perché fra di essi ve ne sono anche di molto preparati.
Non è solo per mancanza di materiale hardware perché in questi anni, nella scuola italiana, sono stati spesi centinaia e centinaia di miliardi per acquisti in tale direzione.
Ma è perché la libertà si paga: usare il software libero richiede più impegno, più fatica, e solo l'insegnante che sente l'oppressione del software proprietario è disposto a fare più fatica per liberarsi dalla schiavitù che questo software impone.
Se gli insegnanti, usando il software proprietario nella didattica, non hanno consapevolezza di essere al servizio degli interessi di un monopolio anziché dei propri studenti e della collettività più in generale, non saranno mai disposti a fare più fatica per liberarsi da questa oppressione.
Dobbiamo allora lavorare sulle coscienze, far prendere coscienza del ruolo sostanzialmente servile che ha un insegnante che, oggi, utilizza il software proprietario monopolistico nella sua didattica.
Solo quando questa consapevolezza si sarà fatta strada tra gli operatori scolastici (insegnanti, tecnici, dirigenti scolastici, dirigenti amministrativi, ecc.) potremo liberarci dal giogo del software proprietario monopolistico nella scuola ed educare gli alunni alla libertà, inserendo l'insegnamento dell'informatica all'interno della moderna cultura scientifica.

Antonio Bernardi

```
biblioteca=>select p.n_inv, l.titolo, p.cod_ut,
                    u.cognome, u.telefono, p.data_p
biblioteca->from prestiti p, utenti u, libri l
biblioteca->where data_r='2050.1.1'  and
biblioteca->      data_p<date('now') and
biblioteca->      p.n_inv=l.n_inv    and
biblioteca->      p.cod_ut=u.cod_ut
biblioteca->;

risultato della select

n_inv | titolo    | cod_ut | cognome  | telefono   | data_p
------+-----------+--------+----------+------------+-----------
1     | La Coppia | A05    | Trevisan | 021-495672 | 2001-01-01
10    | Estasi    | A05    | Trevisan | 021-495672 | 2001-03-21
104   | Il Gufo   | A14    | Bahassou | 07-440118  | 2001-02-22
```

Figura 4

Esempio di select.

Lotta al cybercrime e tutela della privacy: convivenza impossibile?

Da qualche mese è ormai certezza la Convenzione internazionale sul cybercrime: una norma da molti dichiarata indispensabile, soprattutto nei momenti di incertezza che stiamo vivendo, ma da altri fortemente osteggiata. Vediamo perché.

Il 23 novembre scorso, a Budapest, è stata aperta alla firma la Convenzione internazionale sulla criminalità informatica, approvata l'8 novembre dal Comitato dei Ministri europeo. La Convenzione entrerà in vigore quattro mesi dopo la data nella quale il testo risulterà essere stato sottoscritto da cinque Stati, tre dei quali membri del Consiglio d'Europa. Suo scopo è favorire la cooperazione internazionale nella lotta alla criminalità informatica, attraverso l'armonizzazione delle procedure (cioè l'estensione delle stesse regole a tutti gli aderenti) ed il rafforzamento dell'assistenza reciproca tra gli organi giudiziari degli Stati firmatari. Grazie all'accordo le polizie nazionali vedono infatti notevolmente ampliati i loro poteri d'indagine e di azione, e tra di esse viene a crearsi una dinamica rete di collaborazioni. Una organizzazione tale però ha necessariamente bisogno, per lo svolgimento dei suoi compiti, della conservazione, elaborazione e, soprattutto, dello scambio di dati personali. E ciò, nella nostra mente, fa suonare subito un campanello d'allarme: si tratta forse di un'altra delle varie nuove forme di intromissione nelle comunicazioni private, tipo gli Enfopol, Echelon, o il più recente Carnivore (sistema statunitense di intercettazione delle e-mail)? Giustamente la nuova 'criminalità informatica' va contrastata e con mezzi il più possibili penetranti ed efficaci. Ma il guaio è che, come è stato giustamente notato, i veri delinquenti sanno perfettamente come nascondere la propria attività: se non conoscessero al meglio i più moderni mezzi tecnologici e come utilizzarli per il compimento dei loro atti illeciti, non sarebbe certo necessario arrivare all'ideazione di strumenti tanto raffinati per contrastarli. I controlli generalizzati e potenziati verosimilmente non avrebbero altro risultato, quindi, che colpire l'onesto ed ignaro utente di internet. E questo in modo indubbiamente subdolo: senza che se ne renda neanche conto, una parte di sé, di quello che è e della sua libertà, gli verrebbe irrimediabilmente

rubata. E per sempre.

Un'elaborazione sofferta

Fin dal 1997 nel Vecchio Continente si è iniziato ad elaborare un progetto di Convenzione sul cybercrime. Le varie bozze che si sono nel tempo susseguite sono state sempre con forza osteggiate non solo da tecnici, addetti alla protezione e alla sicurezza dei dati, e da giuristi, esperti di diritto alla riservatezza, ma anche da organizzazioni internazionali che si battono per la libertà digitale e per la difesa dei diritti fondamentali di ogni uomo. Risale ad ottobre 2000, ad esempio, la lettera, indirizzata al Consiglio d'Europa dalla Global Internet Liberty Campaign, in cui già venivano espressi forti dubbi sul testo del trattato, sulle sue concrete possibilità d'attuazione e, soprattutto, sulla sua indubbia contrarietà a norme poste a tutela dell'individuo e della sua libertà. In particolare l'organizzazione, che riunisce numerose associazioni europee, sudafricane, australiane e statunitensi, denunciava la discordanza con la Convenzione Europea sui Diritti Umani e la relativa giurisprudenza della Corte Europea per i Diritti Umani. Si opponeva quindi con forza ad ogni disposizione che estendesse in maniera impropria l'autorità di polizia dei governi nazionali oltre i limiti giuridici: ciò avrebbe comportato il rischio di una riduzione degli obblighi del governo di render conto delle sue decisioni e dei suoi atti. Ma anche organismi istituzionali si sono opposti alla Convenzione. Il Gruppo dei Garanti europei ha più volte manifestato, nelle sue riunioni periodiche (conferenza di Stoccolma del 2000 e quella di Atene del '01, le più recenti), forti perplessità in merito. Lo stesso Stefano Rodotà, Garante italiano per la privacy e presidente dei Garanti europei, aveva chiaramente sottolineato come, seppur la lotta alla criminalità informatica e via internet fosse un'indubbia priorità, non si potesse comunque accettare di sacrificare ad essa i diritti del cittadino, inviolabili per principio in un sistema che si definisca democratico. E questo è ciò che invece accadrebbe se le autorità nazionali cedessero alla tentazione di utilizzare

l'enorme quantità di dati oggi facilmente alla portata di tutti per produrre schedature di massa. "Devono essere rigorosamente rispettati – ha detto Rodotà – i principi di finalità e di proporzionalità quando, per combattere la criminalità, si raccolgono informazioni su chi usa Interet." Lotta alla criminalità sì, ma violazione dei principi in materia di intercettazioni delle comunicazioni e di ogni altro principio sancito dalle Convenzioni europee ed internazionali, no.

La Convenzione è ormai realtà

Tra critiche e consensi, la Convenzione sul cybercrime è comunque ormai diventata realtà. Alcune delle osservazioni formulate sono state accolte dal Consiglio d'Europa ed inserite nel testo definitivo; molte altre, però, sono purtroppo rimaste ignorate.
È stato rafforzato il principio del fondamentale rispetto della vita privata, e ciò rendendo esplicito il richiamo all'articolo 8 della Convenzione europea per la salvaguardia dei diritti dell'uomo. Lo Stato a cui è richiesta assistenza da parte di altro Stato firmatario deve così oggi assoggettare il proprio intervento alle norme nazionali interne, che devono essere tali da garantire adeguata tutela ai diritti umani. I provvedimenti adottati in base alla Convenzione devono essere necessari, adeguati e proporzionati alla natura ed alle circostanze del reato.
Alcuni punti però continuano a preoccupare i Garanti europei, le associazioni private e molti singoli cittadini. I fornitori di

servizi Internet, ad esempio, allo scopo di permettere un eventuale controllo da parte delle forze di polizia impegnate nella lotta al cybercrime, devono conservare i dati relativi al traffico in rete (connessione, indirizzo web, ecc.) non più per un periodo indeterminato (come invece si richiedeva nei precedenti disegni). Oggi sono invece obbligati a mantenerli a disposizione degli investigatori 'per almeno 90 giorni'.
Questo, seppur costituisce un miglioramento ed un passo avanti del diritto, non contribuisce di certo a diminuire l'onere, non solo economico, a carico dei provider (per non parlare del fatto che essi, in realtà, non sono altro che dei semplici intermediari nel traffico via web, e spesso non sanno quello che i loro utenti si trasmettono).
Non va poi sottovalutato che la Convenzione, elaborata dal Consiglio d'Europa, si presenta come una normativa internazionale, a cui possono aderire Stati di ogni parte del mondo. Ciò significa che non tutti i Paesi firmatari saranno ugualmente vincolati dalle Convenzioni e direttive che in Europa da decenni tutelano i diritti dell'uomo, anche rispetto all'elaborazione automatica dei dati a carattere personale.

Europol ed Echelon, Gruppo dei Garanti e normativa europea

Europol
Sicuro passo avanti per l'integrazione e la cooperazione tra forze di polizia dei vari Paesi europei, è la recente Convenzione di Bruxelles, del 26 luglio 1996 (entrata in vigore il 1 ottobre '98), che istituisce un ufficio europeo di polizia, l'Europol, la cui creazione era stata già prevista nella Convenzione di Maastricht del '92. Si tratta di un organismo di raccordo tra le polizie nazionali, con compiti essenzialmente informativi ed analitici. Cuore del sistema è quindi un archivio, contenente dati, forniti dalle polizie nazionali, da analizzare e confrontare tra loro allo scopo di migliorare la cooperazione delle forze dell'ordine nella lotta alla criminalità internazionale. Le varie Unità nazionali Europol sono sottoposte a una vigilanza politica e ad una tecnica. In Italia, la prima è affidata ad un Comitato parlamentare di controllo, mentre è compito del Garante della privacy la vigilanza tecnica, finalizzata a che i dati siano trattati in modo lecito e senza pregiudizio dei diritti delle persone.

Echelon
Le agenzie di intelligence di tutto il mondo ne hanno negato testardamente l'esistenza, ma oggi tutti sanno con certezza dell'esistenza di Echelon. Si tratta di una rete (nata nel 1948 in piena Guerra Fredda) estesa a tutto il mondo che indiscriminatamente ascolta e filtra tutto il traffico giornaliero di telefonate, fax ed e-mail; le comunicazioni intercettate vengono sottoposte a controllo casuale e a campione da parte dei Dizionari (computer che cercano ed individuano parole-chiave predefinite), che in questo modo isolano i messaggi sospetti inviandoli alle competenti agenzie per un'analisi più approfondita. Le stazioni di intercettazione dei segnali sono sparse un po' in tutto il mondo, le agenzie interessate sono quella statunitense, inglese, canadese, neozelandese, australiana. Cuore del sistema è negli Stati Uniti, paese che, ormai è dimostrato, ha ricavato indebiti vantaggi di tipo economico o politico dalle informazioni ottenute da Echelon.

Gruppo dei Garanti europei
Il Gruppo per la tutela delle persone con riguardo al trattamento dei dati personali, istituito dall'art. 29 della direttiva 95/46/CE, è composto dai rappresentanti delle autorità di controllo nazionali e da un rappresentante della Commissione europea. A carattere consultivo e indipendente, suo compito è provvedere all'esame delle questioni concernenti l'applicazione nazionale delle direttive, esprimere pareri alla Commissione e formulare raccomandazioni su qualsiasi questione attinente la materia. Il Gruppo si riunisce periodicamente. A tutt'oggi ne è presidente il prof. Stefano Rodotà.

Normativa europea
Convenzione per la salvaguardia dei diritti dell'uomo e delle libertà fondamentali, Roma, 4 novembre 1950; Convenzione per la protezione delle persone in relazione all'elaborazione automatica dei dati a carattere personale, Consiglio d'Europa 28 gennaio 1981 (Convenzione di Strasburgo); Direttiva n. 95/46/CE relativa alla tutela delle persone fisiche con riguardo al trattamento dei dati personali, nonché alla libera circolazione di tali dati, 1995 (attuata in Italia con la legge 675/96).

E' sempre la solita vecchia questione

Forse qualcuno ricorderà la lunga vicenda che ha visto contrapposti Stati Uniti ed Unione Europea circa l'importazione e l'utilizzo commerciale da parte di società americane di dati personali dei cittadini europei.
Dopo aspre polemiche, che hanno fatto temere per il buon esito delle trattative, nella primavera 2000 è stato finalmente raggiunto il tanto sospirato accordo: il c.d. 'Safe Harbor'. In base ad esso i dati personali dei cittadini europei possono essere trasmessi negli Usa, ma lì beneficiano, appunto, di un 'porto sicuro', nel quale il loro trattamento continua ad avvenire secondo le regole UE. L'adesione da parte delle imprese è volontaria, in base al principio dell'autoregolamentazione prevalente negli Stati Uniti. Coloro che vogliono avere rapporti con l'Europa devono chiedere di esser inclusi in un apposito elenco, garantendo di offrire ai consumatori le necessarie tutele di sicurezza. Perché è stato così difficile raggiungere questa intesa? Perché la maggior parte delle imprese statunitensi secondo una recente indagine a tutt'oggi non ha ancora aderito l'accordo? Profonda è la differenza esistente tra le due culture riguardo al modo di intendere la privacy e la tutela della dignità umana; diversa è l'evoluzione storica, e diversa è anche la tradizione giuridica. Gli Usa sono generalmente favorevoli all'unione tra soluzioni di mercato e tutela giuridica mirata a settori: prediligono l'autoregolamentazione, i codici di autodisciplina o di categoria. L'Unione Europea, invece, protegge la riservatezza dei dati attraverso una solida legislazione, e preferisce disporre di un quadro giuridico di riferimento che potenzi il diritto di proprietà dei singoli sui dati personali che li riguardano. Questa vicenda può aiutare a capire la fondatezza dei timori dei Garanti europei circa la reale possibilità di una cooperazione tra polizie ed autorità internazionali priva di regole precise, chiare, restrittive e che garantiscano in pieno i diritti dei cittadini.

Stati Uniti, terrorismo e Patriot Act

C'è poi un altro evento la cui rilevanza e le cui implicazioni non vanno sottovalutate. Il recente attacco terroristico al cuore degli Stati Uniti ha lasciato tutti smarriti. Ma tali drammatici eventi, oltre alle conseguenze belliche sotto gli occhi di tutti, oltreoceano hanno portato anche un'altra conseguenza, meno eclatante, ma di certo ugualmente grave e decisiva. Negli Stati Uniti si è verificato un vero e proprio 'scoppio' di patriottismo, che in alcune sue manifestazioni estreme è a volte apparso esagerato e grottesco a noi italiani, e ci ha anche fatto un po' sorridere. In realtà la cosa è considerata molto molto seria dagli americani. Tanto seria che sulla scia di questo sviscerato amore per la loro Patria la maggior parte dell'opinione pubblica ha sostenuto con entusiasmo una nuova legge, denominata proprio Patriot Act. "Una legislazione attentamente redatta e considerata": così George Bush ha salutato la sua promulgazione. "Una legislazione, ha proseguito il Presidente, che ha incontrato un grande supporto in Parlamento, poiché rispetta e difende le libertà civili garantite dalla Costituzione". Fortunatamente, però, il consenso non è stato poi così unanime. L'unico senatore dissidente e le varie associazioni che si occupano di tutela dei diritti e delle libertà hanno evidenziato come l'eccessiva genericità del testo lasci in pratica all'FBI la possibilità di controllare liberamente un'enorme quantità di informazioni personali, dalle conversazioni telefoniche ai messaggi e-mail, dai passaggi su internet ai dati sanitari e professionali. Tutto ciò senza alcuna tutela giuridica. Ovviamente questo nuovo atteggiamento della politica, e l'appoggio che ha ottenuto, sono dettati dal panico che ha invaso gli Stati Uniti e dall'urgente necessità di combattere il cyber-terrorismo: gli americani (anche se non tutti) si sono dichiarati disposti a sacrificare anche una piccola parte delle propria libertà per il raggiungimento di questo vitale obbiettivo. In realtà, a ben guardare, il sacrificio non è stato poi così minimo ed irrilevante. E già c'è chi inizia parlare anche in Italia di una inderogabile esigenza di maggiori controlli, anche se ciò può portare ad un pregiudizio per la nostra libertà, in nome di un necessario e stringente bisogno di certezza e di sicurezza. Sta nascendo, a scapito degli ignari cittadini del mondo, una rete globale di intercettazioni (sia tecnica che giuridica) che non è facile capire dove potrà portarci: chi ha letto il romanzo "1984" di George Orwell sa di cosa sto parlando. L'importante e almeno sapere il rischio che si corre.

Serena Iovacchini

Ridi, è divertente...

Vuoi Linux con... ?

All'inizio fu VA Research, poi divenne VA Linux e dopo VA Software. Alla fine l'azienda cambiò nome in "VA [niente]" avendo deciso di focalizzare il proprio business sul nulla assoluto. Adesso, dopo i commenti arrabbiati degli investitori, i leader dell'azienda hanno finalmente deciso di cambiare ulteriormente il proprio nome in "VA Burger" ed aprire una catena di ristoranti fast-food in tutta la nazione.

"Dovremo far soldi in qualche modo" ha spiegato Mr. N. Compuhunt, il nuovo vice presidente delle Operazioni Alimentari e Salse Segrete. "Che c'è di meglio che vendere alimentari preprocessati basati su temi geek?"

Negli uffici della dotcom fallita eBuondìMottaByMail.com verrà aperta la prima sede di "VA Burger" in cui sarà disponibile Internet a larga banda su tutti i tavoli, CD-Rom di Linux liberamente distribuibili per tutti i clienti, un pannello led a scritte scorrevoli con le ultime news di Slashdot, e cameriere cozze ma completamente preparate sugli internals del kernel.

Ah... VA Burger venderà cibo, anche. Il menù includerà cose tipo: * Il GNU/Linux/Lattuga/Manzo/Pomodoro/Cetrioli/Cipolle/Ketchup/SalsaSegreta/Pane Burger - 3 centesimi di ciascun GLLMPCCKSPB venduto saranno devoluti al progetto GNU;

* La Sorpresa Fiammante di ESR - Un hamburger talmente pieno di salsa piccante da farvi uscire fiammate dalle narici;

*** burger.rc - un panino completamente configurabile solo per geek hard-core. Questi clienti potranno sedersi ai propri posti, accendere il proprio editor preferito sul terminale del tavolo, e scrivere un complicato file di configurazione per descrivere esattamente quali elementi ottenere nel proprio panino, in che ordine e in quali quantità. Una volta completato potranno spedire quest'ordine via email alle cucine che interpreteranno il file e prepareranno il panino;**

* BurgerXP - Proprio come MS Windows è una simulazione ampiamente imperfetta di un vero sistema operativo così il panino vegetariano è una imperfetta simulazione di vera carne. Nessun animale viene ucciso per la produzione di questo panino, sebbene milioni di assaggiatori beta-sapore-tester hanno riportato ferite al semplice contatto con la miscela.

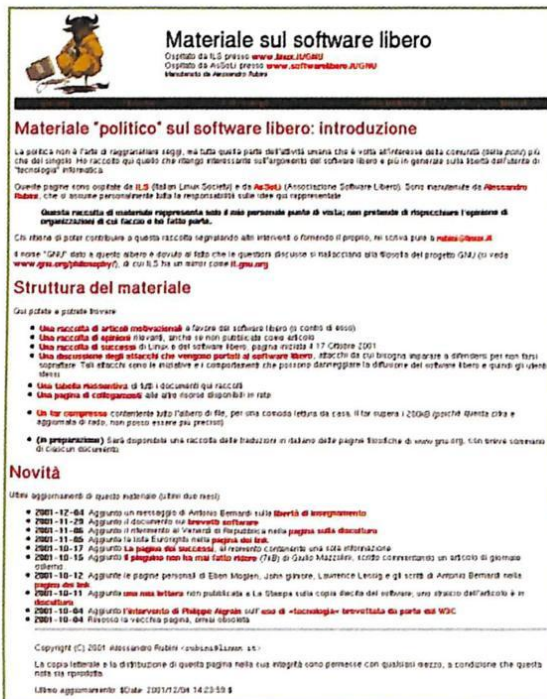
*** L'Hollywood Sandwich (con il DRM Digestion Rights Management(tm) control) - Un panino speciale per i manager e gli avvocati dell'industria hollywoodiana. Ordinando questo panino il cliente deve precedentemente firmare un Accordo di Licenza per il Consumatore Finale che proibisce a più di una persona di consumare lo stesso hamburger, o di consumarlo dopo un certo tempo. Il panino è servito su un piatto con sensori DRM che si accorgono della "condivisione non autorizzata" del piatto (anche detta "pirateria hamburger") e suonano un allarme.**

* Distributore infinito di Caffeina. Ogni tavolo conterrà una porta per un dispositivo CAD (Caffeina ad Accesso Diretto) riempita con caffè caldo o con la speciale miscela EyeOpener(tm) ad alto tasso caffeina. Per solo $2.95 un cliente riceverà un'adattatore d'interfaccia CAD-bocca (volgarmente detta cannuccia) e portà ingerire tutto il liquido che riuscirà ad ingurgitare.

L'azienda spera di guadagnare abbastanza soldi dalla nuova catena di ristoranti per rimanere sul mercato almeno per un altro anno. "Se riusciamo a resistere qualche altro mese almeno possiamo sicuramente battere il record mondiale della più anziana dotcom non fallita" ha spiegato il contabile della VA Burger, con una certa nota di soddisfazione.

Il titolo VA Burger quotato al Nasdaq come *LNUX* e ora quotato nel Nuovo Mercato Rionale come VABU è schizzato di oltre il 50% negli scambi del Venerdì per raggiungere gli 0.01 centesimi, record delle ultime 26 settimane.

da Humorix (James Baughn) Humorix All Linux Humor. All Copied Mottos. All the Time. http://i-want-a-website.com/about-linux/

www.linux.it/GNU

Materiale "politico" sul software libero

E' una raccolta interessantissima di materiale riguardante il software libero che lo stesso curatore Alessandro Rubini, noto esponente della Free Software Foundation, definisce "politico". Il significato di "politica" in questo contesto è opportunamente chiarito nelle prime righe della home page: "quella parte dell'attività umana volta all'interesse della comunità (polis) più che del singolo".

I visitatori non devono quindi aspettarsi di trovare nel sito disquisizioni tecniche, istruzioni sull'uso di programmi e men che meno la possibilità di scaricare software. In compenso sono messi a disposizione una serie di documenti di fondamentale importanza per chi vuole avvicinarsi allo "spirito" e alla "filosofia" del software libero e del progetto GNU in particolare.

Figura 1

La home page del sito.

Organizzazione del Sito

La home page, riportata in Figura, colpisce immediatamente per la sua assoluta semplicità; sono infatti del tutto assenti "effetti speciali", grafici ed altre "carinerie" che appesantiscono e ammorbano tanti siti in giro per la rete. L'unico "vezzo" è l'immagine dello Gnu pifferaio in alto a sinistra, per il resto solo vecchio, sano e significativo testo. La pagina è a tutti gli effetti un contenitore di collegamenti ad altri documenti o a siti che trattano questioni filosofiche, etiche, politiche inerenti il software libero. Si inizia con la barra sotto il titolo che contiene i collegamenti:

- al progetto GNU in lingua italiana;
- alla pagina del progetto GNU riguardante la filosofia del progetto;
- al sito della FSF Europe;
- al sito dell'ASSOLI (ASsociazione SOftware LIbero);
- alla home page della ILS.

Successivamente, dopo alcune precisazioni del curatore riguardanti la paternità delle opinioni contenute nel sito e il motivo del suo nome, troviamo un elenco di altri collegamenti sotto il titolo: Struttura del materiale, seguito dalle Ultime novità, un ulteriore elenco con gli aggiornamenti più recenti (ultimi due mesi). Scorrendo le voci in elenco troviamo nell'ordine:

- una raccolta di articoli motivazionali;
- una raccolta di opinioni;
- una raccolta di successi;
- una raccolta di materiale "pericoloso" per il software libero;
- una pagina di collegamenti ad altre risorse.

Due altre voci dell'elenco sono, a mio parere, molto importanti e meriterebbero forse una collocazione più "in vista":

- la tabella riassuntiva di tutto il materiale contenuto nel sito;
- il tar compresso con l'albero dei file scaricabile per una comoda consultazione offline di tutto il materiale.

Conclusioni

Il sito, che è ospitato presso *www.linux.it/GNU* e presso *www.softwarelibero.it/GNU*, è curato da Alessandro Rubini in modo originale, con una organizzazione ragionata dei contenuti e senza alcuna concessione al superfluo come emerge anche dall'estrema "pulizia" del codice HTML delle varie pagine. Inutili sottolineare, infine, la velocità di accesso alle informazioni, dato il carattere esclusivamente testuale delle stesse.

Fulvio Ferroni